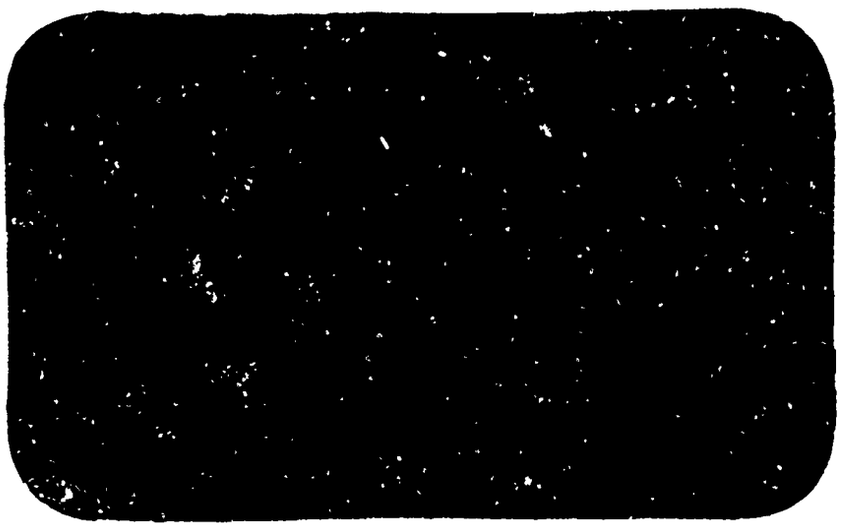


The George Washington University

LOGISTICS RESEARCH PROJECT

Contract N00014-67-A-0214

Task 0001, Project NR 047 001

OFFICE OF NAVAL RESEARCH

DATA INPUT ERROR DETECTION AND CORRECTION PROCEDURES

by

Thomas C. Varley

Serial T-222

2 June 1969

THE GEORGE WASHINGTON UNIVERSITY
Logistics Research Project

Contract N00014-67-A-0214
Task 0001, Project NR 047 001
Office of Naval Research

THE GEORGE WASHINGTON UNIVERSITY
Logistics Research Project

Abstract
of
Serial T-222

DATA INPUT ERROR DETECTION AND CORRECTION PROCEDURES

by

Thomas C. Varley*

This study is an examination of the input data error problem in computerized information systems. The area of concern is the detection and correction of input data errors resulting from human recording during the initial collection of the data.

The knowledge concerning error content in most systems is limited to the hardware components of the system, with little regard for the error content of the input data. Most information system users either (1) assume that all the information presented to them is error free and use the information as presented, or (2) have a complete lack of faith in the information and bypass the system.

Missing from current information system procedures and from the literature is a rationale which provides a basis for intelligent, confident movements toward some middle ground. The provisions of that rationale--the development of a structure of philosophy of the error phenomenon--are the major concern of this paper.

This research attempts to remove some of the mystery surrounding the input error problem. A system for classifying errors by type is developed; attention is paid to the kinds of errors which can be made or introduced at various levels in the data generation-data processing chain. More important, these levels and their potential use to managers and researchers alike provide a conceptual framework in which intelligent discussions concerning the error process can be formulated. The concept of data worth alone provides a significant step forward in building an intelligent detection and correction process.

The basic means of justifying and evaluating automated information systems has been a cost displacement criterion--mainly through reduction of clerical costs. New criteria are being suggested. The basis for the new criteria is associated with the value or worth of the data in the decision process. The basic concepts needed by the decision maker for evaluating the worth of his data are covered in the study. The necessary relationships between data worth, accuracy, and cost are also covered.

The study develops a systematic procedure--a model--for evaluating the various detection and correction alternatives. The final evaluating of the detection and correction procedures to be used in the system is based on cost. This is not displacement cost, but cost associated with improved operations through more accurate information.

The value of information is the worth of the data, and the worth of the data is the data accuracy problem. The major contributor to data accuracy is formal procedures for input error detection and correction. This study has developed these formal procedures.

---

*With the Office of Naval Research.

DATA INPUT ERROR DETECTION AND CORRECTION PROCEDURES

By

Thomas Clair Varley

Bachelor of Arts
The George Washington University, 1960

Master of Arts
The George Washington University, 1962

A Dissertation Submitted to the School of Government and Business Administration of The George Washington University in Partial Fulfillment of the Requirements for the Degree of Doctor of Business Administration

June 1969

Dissertation directed by
Marvin Milton Wofsey
Associate Professor of Management

DATA INPUT ERROR DETECTION AND CORRECTION PROCEDURES

by

Thomas C. Varley

Abstract

This study is an examination of the input data error problem in computerized information systems. The area of concern is the detection and correction of input data errors resulting from human recording during the initial collection of the data.

The knowledge concerning error content in most systems is limited to the hardware components of the system, with little regard for the error content of the input data. Most information system users either (1) assume that all the information presented to them is error free and use the information as presented, or (2) have a complete lack of faith in the information and bypass the system.

Missing from current information system procedures and from the literature is a rationale which provides a basis for intelligent, confident movements toward some middle ground. The provisions of that rationale--the development of a structure of philosophy of the error phenomenon--are the major concern of this paper.

This research attempts to remove some of the mystery surrounding the input error problem. A system for classifying errors by type is developed; attention is paid to the kinds of errors which can be made or introduced at various levels in the data generation-data processing chain. More important, these levels and their potential use to managers and researchers alike provide a conceptual framework in which intelligent discussions concerning the error process can be formulated. The concept of data worth alone provides a significant step forward in building an intelligent detection and correction process.

The basic means of justifying and evaluating automated information systems has been a cost displacement criterion--mainly through reduction of clerical costs. New criteria are being suggested. The basis for the new criteria is associated with the value or worth of the data in the decision process. The basic concepts needed by the decision maker for

evaluating the worth of his data are covered in the study. The necessary relationships between data worth, accuracy, and cost are also covered.

The study develops a systematic procedure--a model--for evaluating the various detection and correction alternatives. The final evaluation of the detection and correction procedures to be used in the system is based on cost. This is not displacement cost, but cost associated with improved operations through more accurate information.

The value of information is the worth of the data, and the worth of the data is the data accuracy problem. The major contributor to data accuracy is formal procedures for input error detection and correction. This study has developed these formal procedures.

PREFACE

This paper is a study of the input data error problem in computerized information systems. It is written primarily for two groups of people. First of all for information system users who wish to see the complexity of providing accurate information. Most information system users accept the results of information systems without questioning the validity of the information. In addition, most users do not realize the various locations where erroneous data can enter the system, or the fundamental procedures for controlling the data collection phases. The further the user is from the initial collection of data, the less likely he is to be associated with the problems of data control.

The paper is also written for research workers, information system designers and system operators to whom it offers a challenge in applying and refining the concepts presented into an operational system. Although not written as a cookbook approach to the detection and correction problem, it is hoped that the paper will be used in conjunction with the development of large computerized information systems.

The idea for this paper was conceived three years ago after the researcher had participated in the design of a large information system for the Department of Navy. The complexity of the error problem and the lack of any real understanding of error creation was realized at this time. Documents on the control of input data elements were nearly non-existent. The use of simple admissibility checks were considered quite adequate at the time. The lack of documentation on error control was not due to poor documentation, but to the fact that data control was not being considered as a problem.

The paper presents a concept for detecting and correcting errors in the input data of a computerized information system. The approach is to develop a conceptual framework for describing in a non-mathematical nature a system for defining, detecting and correcting the input errors.

The purpose of the concept is twofold. First, to give credence to the fact that such a problem exists. Not only does the problem exist, but it is important to the every day operation of large organizations possessing or anticipating the installation of an information system. The second is to build the basic structure for understanding the error problem from the creation of an error to the successful correction or final disposition of the error. The steps between the creation and final disposition are not trivial. Proper selection of the range, depth and location of the detection and correction procedures is expensive, time-consuming, but necessary for a successful information system.

The final form of this paper owes much to the advice, criticism and encouragement of many people. While it is not possible to mention all those who in some way stimulated the author's interest in this problem, several should be mentioned.

First to the committee, Marvin Wofsey, George Allen and Jerome Bracken, sincere thanks for their counsel, suggestions and advice. To colleagues at the Office of Naval Research, the author expresses appreciation for their generous comments and discussions. In particular, my appreciation to Marvin Denicoff who has spent many hours of discussion with me as well as others in the Navy establishment on the importance and need for input accuracy control.

Thanks to Mrs. Eunice Carver for the excellent typing, in many instances done under extreme time constraints.

Finally, to my wife Marby, my deepest thanks. Her editorial assistance made the paper clearer and more readable. Most of all, her constant encouragement has helped through the most difficult parts of this paper.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Purpose

All information systems begin with making observations (sensing) and recording the results. The results are then made available to a user who needs information. The major function of any information system is to provide an accurate picture of the environment in which the system was designed to work. The information needs of the user determines the range of data collected. This means that the range of the data elements collected as well as the depth of detail to which each is collected within the system have importance to the users of the system.[1]

The system manager must have available procedures for detecting and correcting input errors in order to maintain or improve the reliability of the information system. Reliability is important, for if the system users do not believe in the output of the system, the system will be bypassed. The lack of "faith" in data validity has been a major cause for the outright rejection or slow progress of acceptance of automated information systems by today's government, military, and industrial managers. The purpose of the paper is to provide procedures for detecting and correcting data input errors introduced by the human observer.

---

[1]At this point of the paper no attempt will be made to evaluate the importance of the information.

The Data Input Error Problem

Among users and managers of large scale computerized information systems, there is no question of the high incidence of errors in data bases. The failure to isolate, categorize, evaluate, and make logical cost effective corrections or disposition of particular classes of errors have been some of the reasons for the high incidence of errors.

In two large military systems: 1) Air Force Base Level Maintenance System 66-1, and 2) Standard Navy Maintenance and Material Management Information System, the data collected does not pass some of the most elementary admissibility checks. The volume of data created by each of these systems is in excess of 3,000,000 punched cards per month. It has been estimated that the "best" error rate obtainable in these system is five percent. At least five percent of the records will contain a data element that did not pass the systems validation/admissibility checks. This means that about 150,000 records contain simple detectable errors.

It should be emphasized that the current admissibility procedures are designed for and limited to screening for compliance with form completion and field formatting instructions, i.e., left or right justification, alpha vs numeric vs alphanumeric delineation. The statements concerning admissibility do not consider the accuracy of the data element recorded; only that the recorded entry meets the basic characteristics of the element. By basic characteristic is meant a delineation of the data element in structural terms, i.e., maximum number of permissible digits, alpha and numeric designations, acceptability of particular entries in each field position. An example of defining a data element by its basic characteristics might be a stock number which is described as a seven-character, numeric field.

The statement that "stock number" is a seven-digit, numeric field is an example of defining a data element by its basic characteristics. Keeping in mind that current admissibility checks are primarily directed at checking for structural compliance, any entry which satisfies the condition of

seven numeric digits would be counted as a valid number. The limitations of such auditing mechanisms are obvious. Incorrect entries, wearing the proper structural disguise, gain free and easy entry into system data banks.

The previously reported experience of a five percent error rate is related exclusively to admissibility screening. It would not be surprising if the true error rate was an order of magnitude higher.

To look at this problem in another way, Chapdelaine describes a different Air Force system that has repeatedly demonstrated an expected error rate in excess of three percent.[1] The total cost of this system including indirect costs has been estimated at $150,000,000 annually. Therefore, a lower limit on the cost associated with the collection of incorrect data is in excess of $4,500,000 annually. It should be noted that this does not include any cost that might have been incurred in the use of the false data which produced erroneous information on which decisions were based.

The question of errors is by no means limited to military or other government systems. Carlson describes a study which attempted to predict clerical errors in a central bank environment.[2] The error rates were not as high as those described in the above systems. The average for the bank was about 1.2 errors per thousand items checked. Even at this rate, it was considered necessary to improve the method for predicting and detecting errors in the check processing procedure. Of interest were Carlson's remarks about the lack of error detection studies, as well as the routines he developed for the check processing procedure of a bank.

---

[1] P. A. Chapdelaine, Accuracy Control in Source Data Collection. Headquarters, Air Force Logistics Command, Wright-Patterson Air Force Base, Ohio, 1962.

[2] Gary Carlson, "Predicting Clerical Error," Datamation, (February, 1963), pp. 34-36.

Scope of Paper

Why has the error problem not been solved? There seems to be at least two reasons. The first reason is that the whole field of computers and information theory is new. It is new in the sense that only twenty years have passed since the first computers were designed and the classic theory of information was developed by Shannon.[1] The second reason is that, until the very recent past, primary, if not exclusive, emphasis was centered in controlling the errors produced in the hardware (computer).

The concern for improving the internal reliability of the hardware is understandable. However, the emphasis on hardware reliability obscured the necessity for r search related to errors resulting from the data generation process per se. Hardware reliability, frequently sold as a synonym for information system reliability, did not and, logically, could not constitute an acceptable substitute for the design of input error procedures. The solution to these problems could only be accomplished by the employment of techniques tailored to the problems associated with input data error detection and correction. That these two are not synonymous is a bitter lesson that American management has been painfully learning in the current decade.

The Problems

Problems of error effects on the decision process, and the impact that delays have on the process and probability of detection and correction, are in need of study. The cost of detection and correction has not been formally defined. There is, typically, no concept of error priorities in information systems: i.e., the same amount of time is being spent on detecting very low priority errors as on the highest priority errors.

The detection and correction process must make efficient use of available resources. Cost relationships between

---

[1] C. E. Shannon, "A Mathematical Theory of Communications," Bell System Technical Journal, Vol. 27 (October 1948).

the availability and use of ancillary and master files as back-up material, and the concept of error detection and correction at different levels of the organization are all interrelated. It does not seem reasonable to provide complete back-up material at the lowest levels nor to permit the full range of screening activities. In the same way, it does seem possible or desirable to include, at the highest or central computer level, a means for making judgment detection and correction procedures about data elements that are only known at the lowest level.

For example, the use of a master stock record file which contains over a million stock records, if file size were the only consideration, might be considered suitable for duplication at the different audit levels. However, the total cost of such decentralization would also include all the file maintenance required. It is not inconceivable that the file maintenance, which is several thousand records each month, could quickly be more costly than the basic master file generation itself.

The introduction of such a decentralized master file causes many subsidiary problems. One example is the necessity that all master file changes be simultaneously processed at all holding activities to insure uniform auditing of incoming records. The use of master files at decentralized locations does not necessarily lessen the workload of the central activity, but increases the error possibility as well as the time required to process the data.

Degree of Detection and Correction

Additionally, there are questions concerning the degree of detection and correction that should be conducted at the different levels of organization. For example, the lowest level (next to where the source form is generated) suggests that all data validation should be performed there. The rationalization for this point of view is that close proximity to the data generation site permits: (1) tailored processing over a limited segment of the data generating population, (2) deeper knowledge of the underlying processes

which give rise to errors, (3) prompt feedback and direct coordination of errors with the source. Another rationalization is that this concentration and specialization of the auditing function at the lowest levels permits the central computer activity to perform its major tasks of data aggregation and management report generation.

The other side of the coin, however, sees the lower level as utilizing its desired role in the audit process to hide or mask truth. Too much sanitation, too much purification at the lower levels; too fine a filter applied to the original source data eliminate any possibility of the central processing activity uncovering basic trends that are either unique or common to all the decentralized activities. It would seem that there is some middle ground where procedures could be developed to account for such filtering. One such procedure would be for the low level to initiate a particular set of auditing procedures; forwarding to the central processing unit both the original recording, and the corrected recording.

Additionally, there are questions related to the decision process. What effect do errors have on the decision process? Can the decision process use information that is in error so long as the direction of the error is known? What variance or range of error is acceptable in the decision process? How can aggregation help smooth out the errors that would affect the decision process? In attempting to consolidate the problem areas that require research, answers to the following questions will provide the scope of this paper.

The questions are:

(a) What are the criteria for determining auditing responsibilities and procedures to be assigned and employed at the various organizational levels?

(b) If all errors are not detectable, what procedures are available for estimating the "true error rate"?

(c) What are the criteria for determining the extent to which errors should be corrected?

(d) If detected errors cannot be corrected, what is the disposition of the uncorrected, detected errors?

Anticipated Research

It is anticipated that the research will lead to a set of formal procedures to be used by information system designers and managers in creating mechanisms for the detection and correction of data input errors. The research will cover the areas of data worth, error priority, cost of detection, cost of correction, errors related to end use, and the use of statistical and formal procedures in determining data reliability.

Technical Approach

It is anticipated that a survey of a large information system will be conducted. The survey would include a study of the system objectives, and of the current error detection and correction procedures. The survey would also be directed at a review of a sample of the data base used in the system, along with reports and system studies related to error detection and correction.

While the error problem can be divided into three parts: the error detection, the error correction, and the error prevention problem, this paper will be mainly concerned with the detection and correction problem. The prevention problem, while interesting, will be discussed only to the extent required in order to make the paper more meaningful from a system standpoint.

The research will not build programming routines or algorithms, but will describe concepts that can be used in any large information system. It is felt that by describing concepts instead of actual programs, the research would be more general in nature, and would not be tied to any particular hardware system. In the area of cost, the attempt will be to develop criteria for cost-effectiveness evaluation of the error detection and error correction process, but the paper will not build a formal cost-effectiveness model.

The research will require the collection of data reflective of current systems operations to permit a demonstration of how criteria and techniques developed in the paper would apply to real life operating systems. The data will not be used for simulating the effects of the research, but to show how these results could be applied in the current system. The basic system to be studied is a military system which the candidate helped design. Familiarity with this system will simplify the task of identifying and acquiring data essential to the proposed study.

The actual method for development of techniques will employ concepts of error priority, bounding concepts, redundancy, dependent data elements, master file and ancillary file look-ups, cross-reference files, visual check-off lists and templates. The techniques will also draw upon related research from the fields of psychology, electrical engineering, applied statistics, mathematics and information theory. The techniques will be distinguished by the areas where they can be applied, such as at the source of the data, before keypunch, but after source form recording, etc.

The Outline of the Paper

Like Caesar's Gaul, this paper is divided into three parts. The first three chapters provide an introduction; Chapters IV, V, and VI are technical in nature and describe the error detection techniques and the error correction techniques. Chapter VI combines these techniques with costs. In the last part, Chapter VII describes error aggregations, and Chapter VIII outlines the formal check list of procedures that was developed. Finally, Chapter IX provides a summary of the paper and some future recommendations.

While three chapters may seem long as an introduction, each introductory chapter deals with a specific problem or subject. The first chapter is a general introduction to the paper, the next chapters discuss several of the problems associated with the lack of users "faith" in information systems along with definitions of detectable and correctable

errors. Chapter III introduces the reader to some past and current related research. To many readers, Chapter III will be interesting because of a discussion of the scientific disciplines which have contributed to the error detection and correction problem.

CHAPTER II

THE ENVIRONMENT

Information Control and the User

Time, Volume and Location

When the sensor and user of the information are one-and-the-same person, the volume of data to be processed is small, and the time period short, there is little need for a formal information control procedure. However, if the volume increases, or the time is extended to the long run or the sensor and user become separated, then information control procedures are required.

During a short period of time, with a small volume of input, an individual can make rational decisions assuming he has the necessary information. But if either the length of time were extended or the volume of data increased, the capacity of most humans to organize and structure the data into meaningful elements of information would soon be exceeded. The separation of the sensor and user has a similar effect in that procedures are needed between the two that were not required when the sensor and user were the same person. The responsibility for information control belongs both to the system designer, and to the system manager in his capacity as data reliability manager.[1]

---

[1] N. E. Willmorth, System Programming Management, Chapter 13, TM(L)2222/013/01, System Development Corporation, Santa Monica, California, September 1965.

Data Reliability

Data reliability becomes more important to the system user as he is removed from the original source of data. When the user was able to observe and record his own data, there was little question in his mind about the reliability of the data. When he is removed from the data, his faith in the recorded information declines. Davis[1] suggests three reasons for this decline: (1) his control (understanding) of the information has been lessened; (2) he becomes dependent on the computer system with a correspondingly lower level of self-reliance on his part; and (3) his level of assuredness of the accuracy of his decisions decreases in direct proportion to the "levels" he is removed from the basic data.

It is possible to make the opposite point of view based on the fact that the system user would rather not know what is happening internally to the data. However, we must assume that the system user is a rational person interested in making the best possible decisions with as much knowledge about the variables concerning the decision that can be obtained.

Examining the Three Reasons for "Loss of Faith"

Assume that the above three reasons for "loss of faith" are true, and there are no grounds for rejecting them, (in fact, there is evidence to substantiate them).[2] Then it becomes the responsibility of the system designer and manager to provide the means for eliminating these fears of the users. To expand on each of the three reasons for "loss of faith," consider a typical situation where the current manual (non-computerized) information system is computerized.

---

[1]Ruth M. Davis, Information Control in an Information System. Lecture in WORC/TIMS Seminar Series at the Civil Service Building, Washington, D.C., October 1967.

[2]L. R. Flock, Jr., "Seven Deadly Dangers in EDP," Harvard Business Review, (May-June 1962), pp. 91-92.

(a) Understanding of the information: The user's control (understanding) of the information has been lessened. This is by far the most important reason for his fears. While the user may be obtaining the same information as before computerization, it is now presented to him in a strange tabular form, with very little clear text. The computer has replaced the human element in the preparation of the report. There is no one individual with whom the user can communicate to expand his knowledge or understanding of the report.

Furthermore, the user does not have the time to understand how the information was generated. He is curious as to whether the procedures are the same as in the "old" system, or if new algorithms have been developed as part of the computerized information system. To inform the user of the current procedures and to allow the system to include a set of explanatory notes that will be beneficial to the user is part of the overall designer's responsibility.

(b) Dependency on computer results: The dependency on the computer system results in a correspondingly lower level of self-reliance on the part of the user. The problems associated with this fear are twofold. The first fear is associated with a lack of understanding on the part of the user that computer reliability is not synonymous with information reliability. While many hardware manufacturers have sold information systems on the internal reliability of the hardware, it is the responsibility of the system designer to impress the importance of information reliability to the user through appropriate control procedures.

The second part of the fear in this area is that of "computer phobia" -- that which a computer prints out must by association be correct, even though the results are not in line with the experience, judgment, or intuition of the user. This fear is akin to the first area in that the training of the user must take account of his past experience. The information system must be built with enough flexibility to allow the user the opportunity to change his information reports and submit inquiries for further elaboration of the

standard reports. This lack of reliance on one's own judgment and knowledge, initiated by the "computer phobia", must be overcome by the system designer through training and accurate control of the information input requirements and data.

(c) Accuracy of user decisions: The user's assuredness of the accuracy of his decisions decreases in direct proportion to the "levels" he is removed from the information. This cause of "lack of faith" is associated with the first cause; the lack of understanding or control.

Another reason for the user's lack of assurance as to the accuracy of his decisions is due to filtering of information. The information made available to him has passed through various levels and various computer routines. The speed with which such information is presented has increased many fold over the old manual information system, and the manager is forced to make his decisions in a much shorter time interval, so that his output (decisions) can be incorporated into the information system for decision makers at higher levels.

This "need" for faster decisions throughout the organization does not allow the individual user time to assimilate all the information which he may feel is necessary to satisfy his unique decision responsibilities. Because of the lack of time for additional data validation by each decision maker in the decision hierarchy, the question of data accuracy and information control procedures becomes more important.

Responsibilities of the System Designers

It is not the writer's intent to describe the full range of the designers' responsibilities for control. Rather, the interest of this paper is to present a few of the less widely known, but essential, control mechanisms which should be taken into account in the sub-area of user training and acceptance.

In the preceding section, the "lack of faith" was discussed. The outcome of the discussion was to emphasize the basic fact that while automated information systems have succeeded in bringing the user more information more quickly, the user's fears have not been overcome.

To overcome these fears, the designers and managers must maintain a continuous communication channel (feedback loop) with the users. This loop, which entails more than just control, is the sole link connecting the designers and managers to the users. To be effective, the information control section of the feedback loop must, at a minimum, enable the user to:

(1) change formats without disrupting the system;

(2) add or delete data elements from his reports as long as the elements are collected within the system;

(3) prescribe the computations to be performed on his reports;

(4) receive cumulative reports on a periodic schedule;

(5) obtain a narrative on his reports;

(6) change, within reason, the frequency of receipt of the reports;

(7) obtain additional back-up data at a lower level of aggregation;

(8) have a contact point for clarification of his reports;

(9) have an interrogation mode of operation for specific questions.

With the availability of these services provided by the designer, the user would gain confidence in the system's capability for answering his needs. It should, however, be understood that providing each of the above services requires resources. If a user of the system is in need of a unique service not anticipated by the system designer, the user must understand the cost, and be able to justify the increase in cost to an increase in the value of the information content.

Three Classes of Information Systems

In order to discuss error detection and correction, it becomes necessary to establish a general framework of the system structure. The structure, defined by this paper, is

composed of those organizational elements which interact with the data base prior to computerized processing. Such a structure would include the collection sites, the review levels, the source form reduction centers, decentralized local ADP sites, and analytical levels including those at a central processing center.

In general, such a structure applied to three kinds of systems that will be discussed, and to which the error detection and correction procedures will apply. The three basic kinds of computerized information systems can be classified as (1) an on-line real-time information system, (2) a batch processing system with optical scanners for transferring the data from the source form to machine readable data, and (3) the more general batch processing system including source forms and the use of keypunching and verification to convert the data to machine readable form.

On-Line Real-Time Information Systems

The newest and most sophisticated information system from the viewpoint of hardware and system software is the on-line real-time system. The interest in this system is that of its simplicity for data input. In this area, there are only two parts of the total system from a hardware perspective. These are the terminal, which receives and transmits data, and the computer with all of its memory units and system software.

In this system, the sensor inputs new data through the terminals directly to the computer where the system programs are utilized in determining the data accuracy. The input is accepted for use in the system, or it is rejected and returned to the sensor at the terminal for resubmission. As would be expected, almost all of the error detection and correction techniques for this type of system are exercised through computer assisted programs. While the sensor, located at the terminal, has some ability and responsibility for error detection and correction, most of the information control procedures belong to the computer assisted programs.

An example of a functional problem area in which on-line, real-time could be used would include an automatic inventory and warehouse information system. In such a system, the computer would maintain, along with other data, all the records associated with inventory levels, orders (both current and back orders), and inventory item locations.

The system would provide a conversational mode, so that communications between the terminal located in the warehouse and the central computer would be possible. As orders were received by the computer, they would be forwarded to the warehouse for filling. As the orders were filled, the warehouseman would inform the computer via the terminal, and the computer would reduce its quantity on-hand of those items used to fill the order.

The computer would also be programmed to perform its own inventory validation/audit. Assume that during slow periods of the day, the computer starts a systematic inventory audit. This is accomplished by asking the warehouseman to verify the quantity of items located in certain storage locations within the warehouse. The process for such an audit might proceed in the following way:

Computer:	WAREHOUSE NO. 3
Warehouse:	YES
Computer:	WE HAVE NO CURRENT ORDERS TO PROCESS. DO YOU HAVE ANY DELIVERIES WHICH HAVE NOT BEEN STOCKED?
Warehouse:	NO
Computer:	THEN WE NEED TO CHECK SOME STOCK LEVELS.
Warehouse:	WHERE DO YOU WANT TO START?
Computer:	CHECK THE ITEMS IN ROW 4, BIN 3.
Warehouse:	THERE ARE 26 ITEMS.
Computer:	THANK YOU, THAT IS ALSO MY COUNT.
Computer:	CHECK ITEMS IN ROW 5, BIN 4.
Warehouse:	THERE ARE 38 ITEMS.
Computer:	MY COUNT IS NOT 38, WOULD YOU CHECK AGAIN? REMEMBER ROW 5, BIN 4, NOT ROW 4, BIN 5.
Warehouse:	THERE ARE 30 ITEMS.
Computer:	THANK YOU, THAT IS ALSO MY COUNT.

The preceding man-computer interaction used a conversational mode of operation. A computer assisted program determined that a simple error had occurred. It also provided to the sensor, through feedback, information on the cause of his error. While the example that was given is deliberately simple, a more important question might be, what if the warehouseman (sensor) came back to the computer and said that there actually were 38 items in Row 5, Bin 4? The important thing to remember about the on-line real-time system is that the detection is usually provided by the computer, while the correction is provided by the human.

Batch Processing via Optical Scanners

Information systems involving optical scanners are composed of two kinds: those scanners that read by means of mark sensing techniques, and those which have a capability for character readings. The optical scanners' function in the information system is to replace the intermediate transfer device. The raw data or information proceeds from a source form or printed material directly to a machine readable form.

The intent of optical scanning is twofold. The first purpose is to decrease the time required for the raw data or information to be made machine processable. The second purpose is to eliminate the possibilities of errors generated by the intermediate process which the scanner replaced. In general, the intermediate process is composed of TAB equipment; the keypunch and verifier, and a card to tape device.

Examples include the check sorting procedures using special mark sensing devices (MICR)[1] of the banking systems, character scanners used in gasoline credit cards, and charge slips and page readers for library systems. While scanners have been successful in reading the typed and printed information, the acceptability of scanners for handwriting or

[1]Magnetic Ink Character Reader.

handprinting translation has not been demonstrated or applied in any information system now in use.[1,2]

Since the scanners are successful in reading printed and typed material, their use is well justified for those applications where the input is static, i.e., the data are preprinted on the form that is being read by the scanner. While the scanner offers no advantage (at the present time) in detecting, correcting or eliminating errors that were recorded on the source forms, the scanners have eliminated a location for doing error detection and possible correction by making the data "machine readable" in fewer steps. This loss of a detection and correction location outside the hardware system increases the dependence on computer assisted programs for detection and correction.

Batch Processing via Keypunching and Verification

The most common information system is use today employs an additional man-machine interface before the data are available for machine processing. This additional interface is usually composed of a keypunch and verification process which produces an intermediate media of storage.

In using the concept of keypunch and verification, additional information control procedures can be applied before the data are committed to the information system, even though the data are in machine readable form. Since the data are machine readable, the techniques available are those of the computer assisted programs, and, as is true with the optical scanning system, these techniques are not as powerful as those of the on-line real-time system because of the lack of the conversational mode.

---

[1]Recognition Incorporated has a scanner that will read handprinted numerics as long as they are placed in assigned blocks. (See Chapter III, Current Research.)

[2]Bureau of Census has a system called Film Optical Sensing Device for Input to Computers (FOSDIC) which reads microfilm from special forms which have been prepared from the standard forms used in obtaining census of population and housing.

A variation of the common batch processing system should be mentioned. This variation produces, as a by-product of generating the original source form, a punched paper tape in machine readable form. The punched tape is then used in the same manner as cards. An additional advantage is that all data are captured simultaneously with the production of the source document; thereby saving the time of keypunching and verification. While capturing the data in machine readable form at the source eliminates an intermediate step, errors created by the originator of the source form are still possible. The extent of the expected errors can be reduced through error procedures that will be discussed in later chapters.

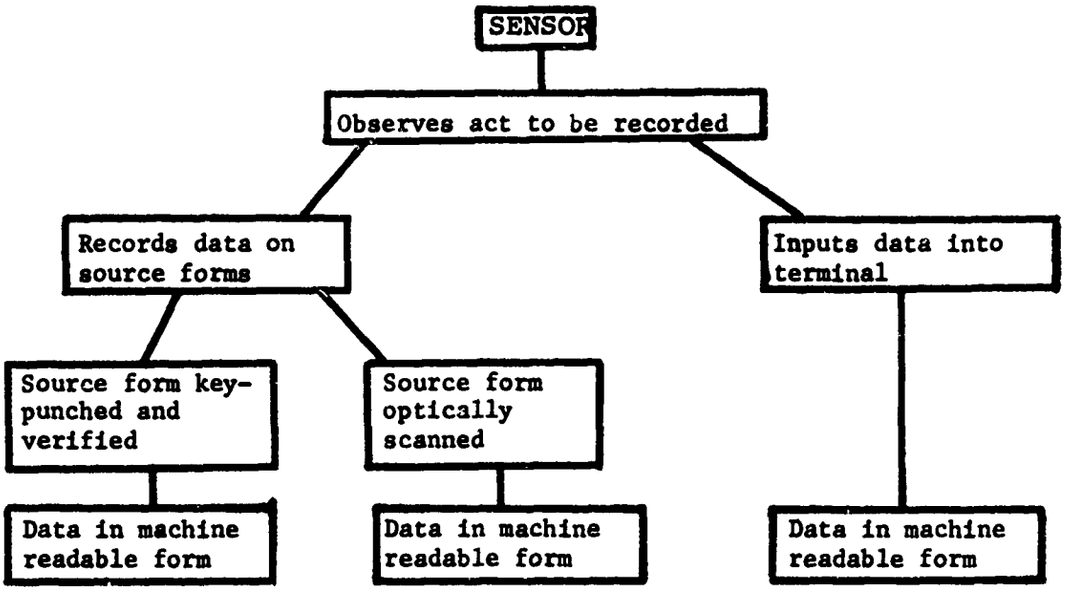
Summary of the Three Kinds of Information Systems

To summarize, the error detection and correction problem is part of the three kinds of information systems discussed. Each is representative of the total classes of information systems available today and in the near future. The procedures for detecting and correcting errors could be different for each kind of system, but, in general, there could be a great amount of overlap. It is not that the logic is different, but the method for implementing the logic varies. In the case of the on-line, real-time system, most of the procedures are computer aided, while in the batch processing via the keypunch and verifier, a great many of the procedures must be manual.

Figure I displays a very aggregated picture of the major components of the three systems. Only those parts that lead to a machine readable form are displayed. Figure I shows that both the optical scanning system and the keypunching system require an additional step even at this high level of aggregation. As the detection and correction procedures are presented, these differences will continue to increase. They will increase in the sense of transfer points where error detection and correction can be performed, and in the complexity of techniques available for error detecting and correction.

Figure I

STEPS OF DATA TRANSFER TO
MACHINE READABLE FORM



How Are Errors Created?

What is an error? How does one know when an error has been found? Are all errors bad, or are some errors more serious than other errors? These are some of the many questions that must be answered by the system designer and manager when he is building and implementing a system. In order to answer the above questions about errors, it is necessary to discuss how errors are created. To discuss the creation of errors, it is necessary to define several assumptions of the error process.

(1) Assume that for a particular data element, the free English text is coded into an alphanumeric code.

(2) Define all acceptable codes of the data element as the Code Set A.

(3) Define for each code in the Code Set A, the code a_i ; for i=1 to n , where n is the number of individual codes in the Code Set A.

(4) Define a_{ij} as the jth character of the ith code in the Code Set A, where i=1 to n and j=1 to m and m is the length of the code necessary to describe all the codes required by the data element.

(5) For standardization, assume for any coded data element that the length of the code is fixed over all codes for that data element.

Given the above definitions and assumptions, the following formal code can be constructed. We have a Code Set A, $\{a_1, a_2, a_3, \ldots, a_i, \ldots, a_n\}$ composed of individual codes which uniquely identify each action or variation needed to describe the data element. Each $a_i \{a_{i1}, a_{i2}, a_{ij}, \ldots, a_{im}\}$ is composed of the individual characters necessary to produce the specific code for each action or variation of the data element. It should be mentioned that for certain kinds of data elements, the previous formal structure does not fully apply. To be specific, such data elements as time, down-time,

equipment operating time; man-hours of work, etc., tend to be of a continuous nature and may not be coded. These data elements do, however, have a code set and observations must belong to the set. But because of their continuous nature, these data elements can have various degrees of accuracy depending on their importance as a data element in the information system. The variation in the degree of accuracy that is allowed will play an important role in the amount of error detection and correction procedures necessary to ensure the required accuracy. This will be discussed later in Chapters IV and V.

In Figure II, the error is defined along with the various actions that he may take in recording an observation.[1] In the language of the preceding section, the sensor first observes an act to record. He may observe the act correctly or incorrectly. Once observed, the sensor records the action. He may record it correctly or incorrectly. Here, he has two alternatives for each of his previous actions. If he observes the act correctly, he selects the appropriate code from the Code Set A, and records this code. In the process of transferring the code to the recording device, the sensor can record the correct code as selected, which belongs to the Code Set A, or the sensor may record an incorrect code which may or may not belong to the Code Set A. In the latter case, where he recorded an incorrect observation, the sensor has committed an error in recording, as opposed to an error in observation, even though the code selected belongs to the Code Set A.

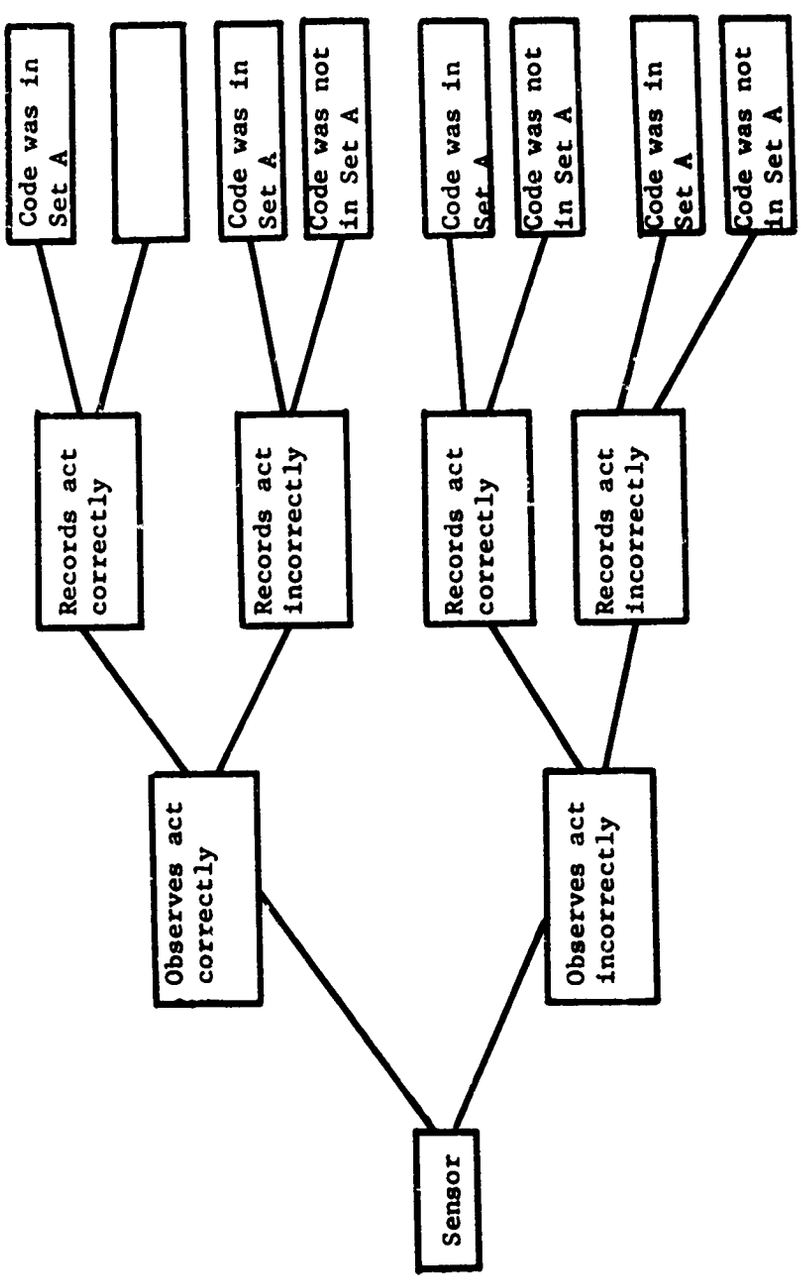
In the case where the sensor observed the act incorrectly, he can record the action as observed, or commit an additional error by recording a code not associated with the code he wished to record. This might be thought of as a compound error.

---

[1] While a sensor is any human or mechanical device that transfers data, it seems more appropriate to describe "it" as a human sensor.

Figure II

RELATIONSHIP BETWEEN SENSOR, AND RECORDED OUTCOME



Type I and Type II Errors

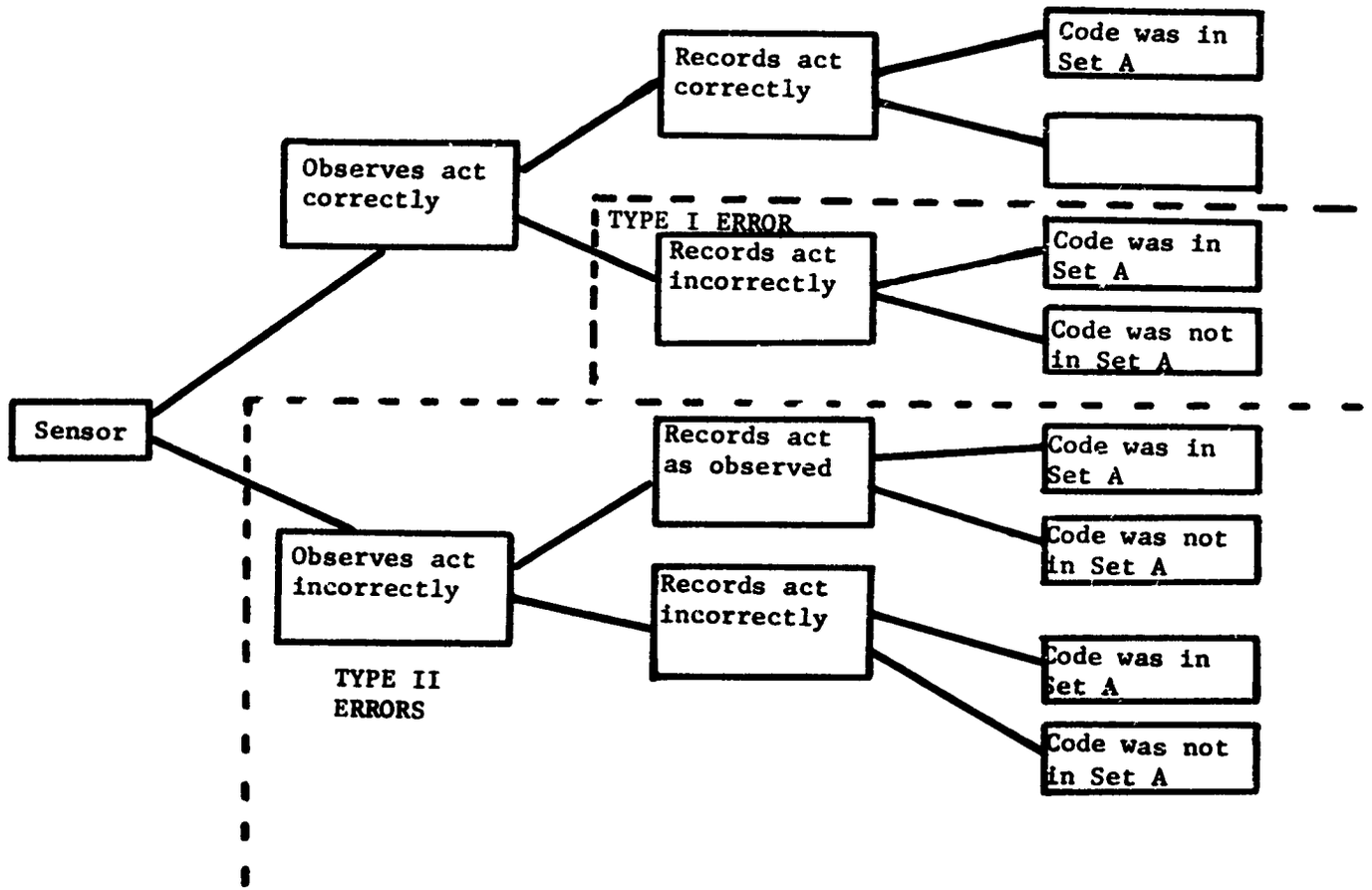
Two kinds of errors can be classified. The first kind of errors, or Type I errors, are errors made after the sensor has observed the correct action, but has recorded the wrong code. This is regardless of whether the code recorded was a member of the Code Set A. The second kind of errors, or Type II errors, are defined as errors where the sensor observed the action incorrectly and recorded that observed code. In addition, the sensor could have committed an additional error by recording another code, again regardless of whether the code finally recorded was a member of the Code Set A.

In Figure III, the diagram is divided to show the error creation process under each of the two kinds of errors. In looking at the end points of Figure III, three separate and distinct situations are described. In the first case, two end points are represented. One end point produces a trivial solution and is meaningless in the error creation process. The other end point leads to the only correctly coded data element. The second case also contains two end points. These two end points belong to the set of errors generated by the Type I error and are errors of recording. The third case contains four end points. These four end points belong to the set of errors generated by the Type II error. It should be noted that one of these Type II Code Set A errors could actually be a correct code. This possibility arises when the sensor observes the act incorrectly, but in recording, accidently records the act that should have been observed originally. Assuming each code is equally likely, the probability of the sensor correcting his error of observation, though the additional error of recording is $P = (\frac{1}{n-1})$, where n is the number of unique codes of Code Set A.

It should be noted that only one path leads to a correctly coded data element, i.e., one out of seven end points. If the probabilities of each outcome were equally likely, over 85 percent of the information collected could contain

Figure III

RECORDED OUTCOME, TYPE I AND TYPE II ERRORS



either a Type I or Type II error. How is the system user protected from such a large possibility of incorrect data?

Detectable and Undetectable Errors

In order for an error to be detectable, it must have one of the following conditions.

(1) It must not be a member of its Code Set.

(2) It is not a member of the subset of other data element Code Sets when taken in specific combination.

Condition (2) is in reality a special case of condition (1), and depends on the question of detecting errors independently within a data element as opposed to detection in conjunction with other data elements. In condition (2) the intersection of two Code Sets provides a new code set for detecting errors belonging to one or the other of the two data elements. However, both of the data elements cannot be corrected; one of the elements must be known or assumed to be correct.

If the code does not meet the two conditions stated, the code is either correct or an undetectable error. In either case, there is no ability to differentiate between the correct code and an undetectable error. As an example of the detection conditions, assume the recording of the stock number of a part that was consumed in the repair of a piece of equipment. To satisfy condition (1), the stock number provided must not be on the official list of stock numbers (not a member of the Code Set) that are in use. If this is the case, then the recorded stock number is in error and this error can be detected, or the admissible list (official list) is in error.[1]

Assume now that the stock number provided was, in fact, a good stock number. That is, the number was on the official list of stock numbers (it is a member of the Code

[1]Assume for this example that the admissible list is correct.

Set). At this point, the error detection process becomes concerned with establishing, not simply the validity of the independent stock number recording, but the validity of its relationship to the equipment identified on the source form.

To make such determinations, the error detection system must both contain and employ lists of all permissible relationships. One such list would be composed of equipment numbers, and the stock numbers of all the parts associated with the repair of each equipment. At the time the stock number was received, the equipment number was also received. The equipment number has been determined to be correct, or at least not a detectable error. By combining the equipment number with the stock number, however, it was determined that the stock number provided was not for one of the parts associated with the equipment.[1]

At this point, an error has been detected: not an error in the stock number itself, but a second order error associated with the application of the stock number. This type of error can be as important to the information users as the first order error.

While an error has been detected, the successful correction cannot be completed. It must first be determined what is in error. Among the still to be determined possibilities, given that the system has detected a "relationship" error, are: (1) an admissible but incorrect stock number has been reported, (2) an admissible but incorrect equipment number has been reported, (3) the master file of permissible relationships is in error.

Correctable and Uncorrectable Errors

In order for an error to be correctable, it first must be detectable; therefore, all undetectable errors are by definition uncorrectable. Additionally, there are detectable errors that are uncorrectable. For a detectable error to be

---

[1]As stated earlier a more important question might be the accuracy of the relationships between the stock numbers and the equipment numbers.

correctable, it must meet at least one of the following conditions:

(1) The code contains error correcting digits that enable unique identification.

(2) The code, when connected to other code sets, establishes unique code set combinations that, through logical progression, are error correcting.

(3) The data element is of such a nature that bounds can be placed on detectable errors.

(4) The coded data element can be returned to the sensor for correction.

(5) The coded data element is of such a nature that statistical techniques (such as past probability estimates) can be used to determine the most likely correct code.

Some data elements will meet more than one of the conditions, which in effect, give alternative ways to correct the data element. In these cases, the least costly alternative should be used if it will give the same degree of correction as a more costly alternative. Other detectable errors will not meet any of the conditions, except possibly condition (4), sending the detectable error back to the sensor for correction. This can be done for any detectable error if the sensor is known. However, sending the error back for correction should be used with discretion. More will be said about this in later chapters.

Several of the other conditions presented are unique in that provisions for their use must be established during initial system design. This includes the concept of check digits which enable correction of one or two or more characters of the code.[1]

---

[1]L. Arquette, L. Calabi and W. E. Hartnett, A Study of Error Correction Codes (Carlisle, Mass.: Parke Mathematical Laboratories, Inc.), Parts I, II, III.

The same is true for multi-field or joint code set correction conditions. Here the relationships are established in such a way that by logical progression a particular data element can be uniquely corrected. This will, in general, call for more than a two-step joint code set relationship. Generally, the techniques of bounds and known statistical procedures will be most helpful to the information system manager who has inherited an on-going system which was built with little or no information control.

As stated earlier, there are detectable errors that are not correctable. For the most part, these uncorrectable detected errors are subsets of the conditions stated for correctability. They include such cases as the complete omission of a data element,[1] judgmental descriptive data elements, and other data elements that are, by their nature, random elements. That is, their bounds are so great that they cannot be placed into any prior distribution with any degree of certainty.

Summary of How Errors are Created

In summary, the process of how errors are created has been defined. The process is general in that each time the data are transferred, the relationship between the sensor and the recorded data has the same decision tree. The end points of the error decision tree become the objectives of the detection and correction process.

The detection process is defined by two basic conditions, either of which is sufficient to detect an error. The correction process is only possible for detected errors, and five conditions are defined as possibilities for correcting the errors. Any one of the five conditions is sufficient for at least partial correction. The degree of correction

---

[1] In cases where the sensor is known, the document could be sent back and the detectable error corrected. This assumes that the sensor has a good recall of the action which generated the document.

required will determine the range to which combination of the correction conditions will be used for any particular data element.

The basic problem that plagues the system designer and manager is the lack of control over the intermediate steps of the error creation process. The only information available to the manager is the recorded data elements, which must be interrogated to determine if the elements were recorded correctly. Since the system designer and manager have only the recorded data elements, they must always start with the assumption that the coded data elements are in error. The error detection procedures are required to prove to their satisfaction or to the degree required by the system objectives that the data elements are correct.

The error creation problem as viewed by the system designer is one where there are two alternatives: either the recorded data element is correct, or it is incorrect. The sensor observes an act and records a result for which the system designer must determine the accuracy.

Summary of Chapter II

Chapter II intended to provide the reader with the environment that faces the system designer and system manager. The environment was provided from both sides of the information system: the side of the user and that of the system designer.

The user side of the environment discussed the lack of faith that many users have of information systems. The use of central procedures and an information feedback loop were described as two ways to explain how these fears could be eliminated.

From the system designer's side, the environment was discussed as the designer looked out at both the user and the sensor or recorder of the data. The responsibilities of the system designers, as observed from the point of view of the user, require the designer to meet the information requirements of all the users.

In attempting to meet the users requirements, the system designer is constrained by the cost of providing the information. A major cost of providing the information is the cost of collecting the necessary data. As the system designer observes the collection of the data, the task of providing accurate and timely information to the user becomes one of the most important and costly tasks. To provide for this task, formal information control procedures are a must.

As a first step in developing such procedures, a definition of how errors are committed was provided. The definition described two major kinds of errors: errors of observation and errors of recording. As a second step in obtaining formal information control, definitions of detectable and correctable errors were developed. The definitions will be used to develop the formal error detection and corrections procedures of Chapters IV and V.

Because of the way errors are created and the lack of information regarding the intermediate steps between the sensor and the recorded data, the system designer and manager must provide techniques to prove that the data are correct to the degree required by the system objectives.

CHAPTER III

RELATED RESEARCH

Introduction

Related research that is directly associated with source data input accuracy problems is almost non-existent. The little that has been accomplished and published is, for the most part, sponsored and supported by the government. There may be studies of industrial systems, but as far as the professional literature is concerned, little is available.

Because of the general lack of directly associated research, the literature suggests other areas which would shed light on the error detection and correction problem. The research areas include those of psychology, electrical engineering, human factors, statistics and mathematics. The following sections will describe some of the areas where the results of these disciplines have helped to provide a partial solution for the error detection and correction problem.

Early Research

Early formal efforts to provide error detection and correction procedures to data input were through the keypunch machine and human verifier. The early use of computers was mainly in the areas of research and scientific computations. In these two areas, the data input was of a small quantity and more effort was placed in problems associated with transferring data within the hardware system than with the problems

of data input.[1] Even today there is quite a large library of literature on error detection and correction within the hardware system. A review of the Journal of the ACM, The IBM Research Reviews, Datamation and Data Processing magazines will substantiate this fact.[2]

The use of the keypuncher for error detection was more of a means than an end. The objective of the keypunch operation was to provide a means for putting data into the computer and not as a means for detecting errors. (Many of the early users of the computer probably did not consider the keypuncher to be a detection device.) However, the introduction of the keypunch operation had twofold results: (1) the detection of errors produced by the originator of the source form by the keypuncher, and (2) the creation of errors by the keypuncher in transferring the data from the source form to the punched card.

The errors that were detected included those of format, omission, and to some extent, legibility errors. The ability to detect legibility errors was possible because of the small volume of input and the close proximity of the keypunch operator to the originator and user of the data. The errors that were created by the operator included those of keypunch shifts, errors of transposition, character permutations, and the insertion of random characters.

Before machine verification of the keypunched document, sight verification was performed by human checkers. The role of the human checker was to match the keypunched document against the original source form in the same manner that a typist would "proofread" a letter or multilith mat before reproduction. The cards with detected errors would then be

---

[1]Jacob Rabinow, "Optical Character Recognition Today," Data Processing Magazine (January, 1966), p. 18.

[2]Aaron Finerman and Lee Rivers, eds., "A Comprehensive Bibliography of Computing Literature, 1967," Association for Computing Machinery.

rekeypunched and again verified to insure correctness of the source forms. Such tasks were time consuming and slow in that each character of the keypunched document had to be read. Although slow, the low volume of input prevented the operation from becoming a problem.

With the advent of the machine verifier, the verification speed was increased to that obtainable by a keypunch operator. As a result of the verification process now performed by the machine, many felt that the data input accuracy problem had been solved. However, in truth, the verifier had, as its chief duty, a complete (100 percent) review of all the characters of a previously punched card, i.e., the automation of a completely manual system -- the human check.

The objective was that two different operators would insure that the data presented on the source form were, in fact, punched and correct. Disagreements between the keypunch operator and the verifier would be tagged and retained for clarification by the source document originator.

The result of a study by Klemmer and Lockhead[1] showed that for four of twenty keypunch installations surveyed, the range of verification errors caught lay between 1/1600 and 1/4300 keystrokes. This could be interpreted to mean that there is a small chance of correcting errors created by the keypuncher. Another interpretation is that the keypuncher does not create many errors. However, in either case the excess time, machinery and people required for verification was not cost-effective.

The low amount of verification errors reported does not suggest that the source forms are error free. The function of the verifier is to duplicate the keypunch operation, stroke for stroke, from the source form, not to interpolate the data recorded on the source form. Clerical errors do result from keypunching, and several studies suggest that

---

[1]E. T. Klemmer and G. R. Lockhead, "Productivity and Errors in Two Keying Tasks - A Field Study," J. Applied Psychology, 43 (1963), pp. 401-408.

misspelling and keypunch shifts are the major causes of the errors in certain systems.

In another study Owsowitz and Sweetland[1] showed that a large error rate was generated because the keypuncher had problems in legibility of coded data letter pattern familiarity. They suggested that the keypuncher be provided with a list of codes. The list would be used any time a problem arose as to the legibility of a coded element that the keypuncher could not readily decipher. The experiment showed that errors did decrease and that learning did take place for the keypuncher. The problem seems to reflect the ability of the keypunch operator to understand what is expected of the data toward the objectives of the system.

Research related to the keypunch operator is not completely independent of research related to the originator of the source form. The area of human factors has contributed through its research on form design, including the color, type size, paper size, logical layout and spacing. These factors are just as important to successful keypunching as to successful form completion, yet they do not in themselves add directly to the error detection and correction process.

Additionally, there has been research in human memory: specifically, in immediate recall. This latter research has added to a better understanding of both the originator and the keypuncher. Chapdelaine[2] describes an experiment to test the error rate for transferring coded data from one form to another. The codes varied from five to fifteen digits in length. The results showed that certain digit lengths had a higher error rate than others. In particular, the ordering was 5,6,7,8,10,11,13,9,14,12, and 15. In fact, if a 12-digit code was required, the error rate could be reduced by 50 percent if a 13-digit code was used in its place.

---

[1]S. Owsowitz and A. Sweetland, Factors Affecting Coding Errors. Rand Memorandum RM-4346-PR. (Santa Monica, Calif.: The Rand Corporation, 1965).

[2]Chapdelaine, Control in Data Collection, p. 15.

The results of the Chapdelaine study were consistent with earlier studies performed by Wechsler, Crannell and Parrish on immediate recall of numeric information through oral presentation and oral recall.[1,2] The studies showed that for immediate recall the span was slightly better than seven digits.

Similarly, Conrad[3] in a study utilizing telephone operators, who were required to memorize 8-, 9-, and 10-digit numbers, and record them, found that for the 8-digit numbers, 30 percent were in error; for the 9-digit, 44 percent were in error; and for the 10-digit, 54 percent were in error. This suggests that in general the shorter digit codes are more easily retained. Probably the most popular research in this area is Miller's "The Magical Number Seven, Plus or Minus Two."[4] This report describes the human capacity for recall on digits.

The research on immediate recall provides direction and consideration to the information system designer and manager. First, it describes what should be taken into consideration in designing codes, and secondly, it describes where effort should be placed in considering error detection and correction techniques based on the expected error rate of different codes.

Owsowitz and Sweetland[5] took a slightly different tack to error detection and correction, by looking for factors that contribute to coding errors. They conducted a

---

[1]David Wechsler, The Measurement of Adult Intelligence (Baltimore, Md.: The Williams and Wilkins Company, 1944), pp. 83-85.

[2]C. W. Crannell and J. M. Parrish, "A Comparison of Immediate Memory Span for Digits, Letters and Words," The Journal of Psychology, 40 (1957), pp. 319-327.

[3]R. Conrad, "Errors of Immediate Memory," The British Journal of Psychology, 50(4) (November, 1959), pp. 349-359.

[4]George A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capability for Processing Information," The Psychological Review, 63(2) (March, 1956), pp. 81-97.

[5]Owsowitz and Sweetland, Factors Affecting Errors.

series of experiments completely within a three (3) character code that was composed of the various combinations of alpha and numeric characters. Of their findings, two seem most important:

(1) Coding errors are proportional to the alpha content. Numeric does have the smallest error rates and as the alpha content increases, so does the error rate.

(2) Errors committed with mixed codes reveal a position effect. Codes with the alpha-numeric-alpha sequence, and the converse numeric-alpha-numeric, have higher error rates than do codes without this "odd man in the middle" construction.

Again, such results tend to develop a picture that error correction and detection procedures should be based on expected error content as well as error priority.

Current Research

The related current research can be classified into three major areas. The first area is that associated with character recognition, the second area is associated with automatic source data collection, and the third is linked with on-line computer systems.

Character Recognition

The relationship between research originating in the field of optical character recognition and that of error detection and correction has been minimal. This is especially true for the area of source form recording, for the scanning system replaces the conventional keypunch and verifier. The current optical scanners interface with the information system at a higher level than is required to be of greatest benefit to source data accuracy.

The advantages of optical character recognition should, however, be realized and applied to the information system where possible. One such advantage is the transfer of raw

data into machine readable form much more quickly than can be performed by the keypunch procedure. However, with the current state-of-the-art of optical scanners, their use in management information systems is limited to those systems that produce fixed data in a fixed format.

There are two major problems that must be solved before the optical scanner can be considered practical for management information systems. The first is the ability to recognize the specific font that is being used. The second is to recognize accurately the handwriting, both printing[1] and script.[2] The problem of multi-font is to recognize which font set is being used since several fonts have similar looking characters and the area available for discrimination becomes quite small.

Research in the area of handwriting is progressing in two ways. The printing route -- where the writer is required to print within a certain limited box -- and the longhand route. Progress is being made in the print research since the biggest problem in script writing is the lack of uniformity in the spacing between letters.

Source Data Automation

Related research in the area of source data automation is divided into three areas. The first area is associated with pre-punched forms and receipts. The second area is associated with special recording devices, tags, plates, cards, etc., on which data are either pre-punched or engraved. The third area is through special attachments to standard typewriters or recording devices.

---

[1]Recognition Equipment Company does have an optical character reader for handwriting of numeric if the numerics are placed in predescribed blocks and written carefully.

[2]IBM has an experimental model for handwriting.

Pre-punched forms and receipts

Techniques under this area are generally of two kinds: the "tub-file" process and the "turn-around" process. In utilizing the "tub-file" process, pre-punched cards are the usual source of entry into the system. The cards are located with the end item at the source of use, and as the item is dispersed, the pre-punched card is entered into the system.

The "turn-around" process is one that has been highly successful in the utility field. It is comprised of a two-part form: one part, the bill and a second part which is a pre-punched receipt to be returned to the utility with remittance. The pre-punched portion includes the account number, the amount of the bill and other accounting information of interest to the utility.

When the receipt arrives at the utility, the system allows for positive human scanning of the amount of the bill and the amount paid. If these match, no further human intervention is required; if they do not match, the checker makes note of the amount paid and this is written on the receipt stub. The stub is then forwarded through the data processing chain where the marked stub is read and put in machine readable form for processing. If the amount paid was less than the bill, the account is only given credit for the amount paid; if the amount paid is more than the bill, the account is given credit for the difference between the bill and the amount paid.

Special machines, tags, plates and cards

The techniques of this area are similar to those above except that in this category special stations or collection stations are set up for putting in the basic data. The tags and cards are more permanent in nature and can either be carried by a person or located at the special equipment for use by the operators or data collector. Applications of this are the airline ticket reservation systems, railroad ticket reservations, central warehouse operations, gasoline credit cards, etc.

In both of the types of automatic collection process, the data must be fairly static in the sense that very little changes over time as to what is pre-punched or coded on cards. The location and number of sensors who provide the information or hold the cards can be large, and, in fact, the credit card field is very large. Yet, the amount of information which must be scanned or punched compared to the total data content of the record is small.

Special machines -- by-product

The third area of automatic source data collection is associated with attaching collection devices on standard equipment, such as a typewriter or other standard recording device. The approach of such a procedure produces punched paper tape as a by-product of the original operation. The production of the paper tape is then used for all future operations requiring data processing. Examples of such an operation include the processing of data from a sales invoice. The data are presented to the typist who types the original order. At the same time a paper tape is produced which contains all the accounting, distribution, and billing data needed to process the order.

The advantages of using automatic source data collection are mainly in the area of few keystrokes, which means fewer people handling the data, thus fewer chances of errors. This also includes fewer people associated with the collection process. The detection and correction process can be greatly enhanced through the use of such devices in the collection of data.

It should be pointed out that there are disadvantages associated with automatic source data collection. Perhaps the biggest disadvantage is the data structure which must be fairly constant as to field size and elements collected.

<u>On-Line Real-Time Systems</u>

Probably the most important advances, as far as the user is concerned, are the man-machine system interfaces.

The computer industry's history of eliminating the human from the system, i.e., the early automation theme, has now been replaced by a more modern theme of man-machine interaction.

The object of the modern theme is to get the man back into the system. How is this to be accomplished? The current practice is through the new technology of on-line information systems. The fears of the user are to be reduced by the ability of the machine (computer) and man to converse with each other. The key to this communication is both hardware design and system software development.

Many books have been written on the subject of hardware components as well as in the general area of software.[1] We will not discuss specific hardware systems, but will discuss the way in which these on-line systems are working to provide better data through information control.

Both of the areas of data input and information output are greatly helped by the on-line process. In the area of information output, the user has, through the means of exception and interrogation modes, the ability to prescribe the content, form and timing of the response or output. In this manner, the user is better able to describe the data required for his information through the man-computer interplay. The only requirement is that the data required be part of the data collected.

Such a procedure differs considerably from the batch processing method in common use today. The batch method gives the user all the information at the same time and at the same level. Even if the information were correct, there would be many cases where the user was not in a position to analyze all the data, or at that particular time, was not interested in the information.

In the area of data input, the user has the potential of placing his own input directly in the central processing

---

[1]James Martin, The Design of Real-time Computer Systems, 2nd ed. (New York: McGraw-Hill Book Company, 1967).

unit by means of a remote terminal. Additionally, the software developments allow the user to "see" exactly where his data were placed in relationship to other data in the file. This is accomplished by presenting to him the record immediately preceding his input and the record immediately following his input.

The response is displayed through his console which is equipped with either a CRT (Cathode Ray Tube) or typewriter. The use of such a display response procedure will help to eliminate the user's fear of the data not being included, lost or misfiled in the data bank. Additionally, there are procedures which allow the sensor to screen the data input on a CRT display before the data are transmitted to the computer for further validation.

With the development of the conversational mode, and on-line real-time inquiry capability, the computer potential for the detection and correction of input errors is greatly increased. Even if the detection capabilities remain the same, the ability to use more sophisticated computer aided programs will reduce the time and cost of detection.

In the area of correction, one of the current methods in use, namely that of returning the detected error to the sensor, will gain more importance. The detected error can be submitted quickly (in a matter of seconds) to the sensor for correction, before the sensor leaves the terminal, or, more importantly, before the sensor forgets what data were submitted.

While there are other input devices in addition to that of a keyboard terminal for use with on-line real-time systems, the keyboard is by far the most versatile in its ability to capture all of the different kinds of data usually generated in a management-information system.

Future Trends

Optical Character Recognition Devices

The future use of Optical Character Recognition (OCR) during the next decade will produce great cost saving for

certain areas. The use of multi-font machines is just coming into being, and will be perfected within the next few years.

Research in handwriting is progressing. There is currently one machine that will read printed numerics if the numerics are placed within specific blocks, and an experimental handwriting reader is in operation. Future research directed towards perfecting the reading of either longhand or printing will provide useful long run economics. The new technology will require a lower cost per unit of information before it becomes acceptable.

Communications

The requirements for communications will increase during the next decade. The telephone companies have estimated that during the early 1970's, more than 50 percent of their communication will be the flow of data rather than the flow of voice communications. Future trends are to increase the speed of transmitting to a level comparable to machine speed. Such increased speeds will require the use of wide band microwave communication nets instead of land line use for higher comparable rates. Current microwave and broad band telephone facilities are transmitting at speeds up to 50,000 characters per second.[1]

Included in the communication research are the interface problems of equipment. There is a growing requirement that equipment must be designed to overcome these problems by providing some minimum number of standard modules.

An emerging technology still in research, yet exciting enough to be considered is that of lasers. The use of the high energy content of a laser beam to write alphanumeric or graphical data as a source of display information is under investigation. The character writing speeds of 100,000 characters per second is within the realm of possibilities within the next decade.

---

[1]Martin, Real-time Computer Systems, p. 280.

In the area of on-line, real-time, progress will move to the time-shared information system. Such future systems will be completely integrated so that data flow will be between computers as well as between terminals for direct end use. The data input procedures will be combinations of automatic source data techniques for data of a static nature, while more dynamic data will be put in the system by means of terminals utilizing keyboards, light pen and voice character recognition equipment.

Summary

Chapter III intended to provide the reader with some of the relevant research, past, present and future that has or will affect the error problem. In general the research has not been directed to the data input problem, but has been performed for other purposes.

Early research can be associated with keypunch machinery and human verification. The primary objective was associated with transferring data to the hardware system rather than with data input problems. With the advent of the machine verifier, verification speed was increased to that obtainable by a keypunch operator.

Research related to the machine verification has shown that very little errors are found by the process. The average range of verification errors caught lay between 1/1600 and 1/4300 keystrokes.

Current research is related to character recognition, automatic source data collection, and on-line computer systems. Character recognition has advanced mainly in the banking industry through standardization of MICR. The advances in optical scanning of printed and script are progressing. However, technical problems concerning multi-font recognition and character spacing are still major problems. Source data collection techniques are providing quicker ways to transfer raw data into machine readable form via a tape media. However, significant advances in data input error control have been neglected in lieu of eliminating transfer errors after initial recording and elimination of duplication of processing.

CHAPTER IV

ERROR DETECTION PROCEDURES

Introduction

The concept of error detection is to provide, through well defined procedures, methodology and computer aided programs, the ability for detecting the introduction or presence of erroneous data in the information system.

Error detection is tied closely to the place where data either enter the system or are transferred within the system. The actual number of detection locations as well as the number of levels at each detection location are a function of the size and kind of information system being developed or in use. The more the information system is like a conventional batch processing system, the more transfer and data generating locations it possesses.

In deciding what error detection procedures to employ, several constraints should be considered. The most important constraints are those dictated by the system objectives. These include such requirements as the data elements to be collected, their source, other general characteristics of the data, and finally the end uses to which they are applicable. With these and similar constraints in mind it seems necessary to describe the what (object), where (location), and why (reason) for error detection before discussing the how (procedures).

The next three sections define and describe the what, where and why of error detection procedures. In the next section the what (object) is concerned with defining a concept of classes of data, their characteristics and the role they play in the error detection process. In the following

section, the where (location) is defined and described in terms of error detection locations. The locations are genetic in nature and form a complete set from data generator to information user. The last of the three sections describes the why or reason for error detection. Here the why refers to the selection of error detection procedures at the various detection locations. The final section of the chapter describes the procedures for detecting errors at the various detection locations.

Classes of Data

In describing information system error detection procedures, the data elements that are to be subjected to the techniques are generally of several kinds. A classification of all the data collected in information systems would provide a general framework into which error detection procedures could be developed. Such a classification would have several benefits. Two such benefits are: (1) the discussion of error detection techniques within the classes as well as between data classes and (2) a clearer understanding of the advantages and disadvantages of the error detection techniques as they relate to the classes of data.

Static Data

Static data elements are those data elements that have a fixed code set and fixed relationships between codes and end items. That is, the code is constant for a particular end item and the code does not change as long as the end item does not change. Examples of static data elements are names, social security numbers, bank account numbers, manufacturers part numbers, equipment serial numbers, a geographic location, an individuals' sex, and a badge number to name just a few.

The concept of static data elements has meaning in that once an individual, equipment, part or other end item is allowed to enter the system, it possesses a unique code which belongs to no one else. In addition, if the item

leaves the system, the code is retired for a respectable period of time. For some static end items, there is only one such end item and it can only be recorded from one particular place, such as a serialized machine, while for other static data elements there could be many copies of the same end item in the system, all having the same code.

Static data elements have several advantages in an information system that are not possible with other classes of data.

(1) Automatic input

The static data elements are highly accessible to automatic source data collection devices. In many cases the static data can be pre-coded on a plastic card, such as gasoline and department store credit cards, pre-punched in a punch card, or automatically set in a keypunch machine program card. In cases where the information system uses terminals as a means of putting data in the system, the terminal can be pre-coded to include static data concerning its location, user, etc.

(2) Less detection required

If static data are pre-coded and submitted through an automatic source data collection device, the error detection procedures are simpler, as far as detecting an error by means of belonging to the code set.

The error detection problem for such data is associated more with selecting the proper pre-coded plate or card than with detecting errors internal to the code such as transposition errors. Such an error is possible since many plates could be available to the same data recorder. The use of automatic collection devices changes the techniques as well as the locations for error detection and correction.

(3) Use of static data elements in dependency checks

Because static data are to a great extent error free, in the sense of errors in recording, they can be used as dependent checks on other data elements that are not as

static. Such joint code sets could involve cross-reference files between the static data elements, and more dynamic data elements, which would be used for detecting errors in the more dynamic data elements. The concept of a static data element does not mean that the population of the data element is constant or unable to change. Through the addition and deletion of new items, the population is changing which causes the file of the static elements to be somewhat dynamic.

Dynamic Data

The concept of static data elements considers those individual data elements that have a very small probability of change during the information system life. The concept of dynamic data elements considers those data elements that by their nature change frequently during the life of the information system. The frequency of change is intended to mean that the data element has a wide range of codes that can be selected for any particular action.

While the code set is well defined, a data generator may observe and record within a short time a different dynamic code for several actions that were observed. In these same actions the static data elements recorded could be the same.

Such a variation in allowable codes requires additional error detection procedures than were required for static data. The dynamic data elements require error detection procedures in the data recording phase as well as in the data observation phase. Examples of dynamic data elements include the number of man-hours to complete a job, the clock time a job started or stopped, and the cost of a job from a maintenance or job-shop management system. In addition, in an inventory system, such data elements as stock on hand, number of demands for items, and frequency of demands would be dynamic data elements.

The degree of error detection at the lower system locations depends on the amount of computer-assisted-programs

and master files that are available at that location. The reason for such equipment and master files is that statistical techniques seem to be the best detection device for determining the bounds or limits that can be placed on such data elements. In addition a fairly large data base is needed to provide the statistical validity needed for the procedures. It is quite possible that different geographic areas would require different limits for the same data element. Such stratification would require both a statistical national norm and a regional norm. Having regional norms increases the error detection procedures needed to insure the degree of accuracy and precision required by the systems specifications.

Factual and Judgmental Data

Within the two main classes of static and dynamic data, a second classification is possible. While the second classification is a refinement of the first, it has several advantages that should be helpful in the development of error detection procedures.

Factual data

The concept of factual data elements refers to data elements that convey an actual and known fact. If a data element is to be considered factual, the method of observing and recording the data element is well known. In addition, the element is taken from an accepted publication which has been authenticated by the system procedures.

Factual data elements are found in both the static and dynamic data classes. In the static class, the factual elements are those that are unique to a source location. Examples are equipment serial numbers, name of organization, activity or division, an individual's name, address, or work station. Other examples include those static data elements that are common among many source locations such as manufacturers part numbers, bank account numbers, standard unit of issue of materials, equipments installed, etc.

In the dynamic data class, factual data elements are those that have a continuous code set such as time, dates, actual cost and observed man-hours.

Judgmental data

The concept of judgmental data elements is associated with the estimates people make in observing and recording the particular code. In the first case, the estimate is placed with the data generator in selecting the proper code. In the second case the data recorder only records that which was estimated by others and presented to him for recording. As with the factual data elements, the judgmental elements are found in both the static and dynamic data classes.

For the static class the judgmental data are of the second kind. That is, where the data available for recording were themselves estimated by someone other than the data generator.

Examples of static judgmental data include engineering or physical characteristics of end items such as weight and size (dimensions) of components, measurement ton equivalent and similar data elements. It would generally be expected that characteristics such as weight would be static for a particular component for all of its location. However, as is generally observed, most weight is estimated for fairly large items as well as many smaller items. While such estimates may be accurate in the aggregate, such as those used in military planning factors, or standard job costs, any individual estimate may be out of acceptable limits for use in the information system.

In the dynamic data class, judgmental data elements are those elements that have the property of alternative codes. This allows the data recorder, on the basis of his own judgment, to select the one code that best describes the situation. Examples include such elements as the primary and secondary cause of equipment failure, and estimates of man-hours and costs.

The development of error detection procedures for such data elements must consider the wide variation that is possible. In view of these variations, statistical procedures seem to be of prime importance. This does not mean that code set relationships will not be used, but the independent testing of such data elements tends to fall in the statistical inference domain. While many of the dynamic judgment data elements will have a wide variation, it does not seem reasonable to expect the same wide variation from static judgment data elements. This can be related to the fact that the static judgmental elements are characterized by end items, which by their nature should not have a wide variation.

Environment, Descriptive and Action Data

In the last two sections, classes of data were described by (1) the characteristics of their probability of change and (2) by their accuracy in observation or recording. In this section a third characteristic will be developed. This characteristic is associated with the kind of information the data element possesses. While factual and judgmental data elements were described in terms of being associated with static and dynamic classes, this third characteristic will be described across the static and dynamic classes.

Figure IV shows a matrix of the three-way classification of the data classes.

FIGURE IV

CLASSIFICATION MATRIX OF DATA CLASSES

	Static		Dynamic	
	Factual	Judgmental	Factual	Judgmental
Environmental				
Descriptive				
Action				

Environmental data

Environmental data are defined as those data elements that report the condition or context in which the recorded action is taking place. That is, these data elements tell where and when the action is being recorded. Such data elements would include an organizational code, a location, an address, the date of the action, and the specific time of the action.

The use of the environmental data elements in the error detection process has properties that range from those of static factual to those of dynamic judgmental. Because of such a range, the knowledge that they are environmental provides a relationship that can be useful in the error detection process.

For example, consider an equipment serial number (static factual data element), the date of the action (dynamic-factual) and the condition of the equipment (dynamic judgmental). These can be related through statements concerning where and when the action occurred in relation to other actions involving the same specific equipment or the same equipment in all of its different environments.

Descriptive data

Descriptive data are defined as those data elements that relate to a further delineation of the characteristics of the recorded action. That is, the descriptive data help to define the objective of the action. As an example, consider the following situations. Individuals with the same surname are further defined by first names, initials and place of residence or business and occupation. An electric motor which can perform in many applications can be distinguished by data elements which reflect these different applications.

Descriptive data, as defined, provides the key to the recording of the data, and as the object of the action, the error detection procedures to be developed are of prime

importance, and require a considerable detection procedure at all detection locations.

Action data

Action data are defined as those elements that describe the why and how concerning the object of the recorded action. Such action data elements explain to the higher echelons decisions taken at the location of the action or the use of organizational procedures that guide the way operations are performed. Examples of action data elements would include reason for starting the action, reason for stopping the action, methods used in completing the action, and resources consumed. The above data elements are examples from a job-shop information system.

The error detection procedures associated with this class of data elements encompass code set relationships as well as statistical inference techniques. The majority of the techniques would be of a statistical nature at the higher detection locations, while the lower levels would include both statistical and relationship techniques.

Error Detection Locations

Inherent in the design of any information system is the requirement for some form of data audit. The data audit can be informal or formal. The more formal such an audit becomes, the more complicated the procedures and techniques required for the error detection phase of the audit. While information systems differ, there seems to be at most seven independent generic error detection locations. Not all information systems will contain all seven locations, but all locations that a system has will be contained within the seven.

The Data Generator

The basic function of the data generators is to provide data to the system. While there may be formal procedures for entering the data, these should not be considered

as error detection procedures. The detection procedures apply only after the generator has recorded data to be submitted to the information system. At that time procedures could be presented; such procedures are structured or permissive in nature. Examples of structured procedures include requirements for conducting defined and enforced tests or matches against the recorded data as predetermined times. On the other hand, permissive procedures would include a statement which indicates that the data generator should review the recorded data elements before submitting the form to his supervisor.

If the procedures are structured, they become part of a internal data audit, and place formal requirements on the generator. If the procedures are permissive in nature, there is not formal internal audit at data generation.

Data Checker

The basic function of a data checker is to validate the data prepared by the data generator. Such a data checker is usually located with the data generator and can perform the data checking function as a secondary duty. Examples of data checkers are co-workers, supervisors, stock clerks, cashiers and scorekeepers. While the detection function may be a secondary duty to the data checker, formal procedures can be developed at this location.

Keypunch Location

Many would consider the keypuncher as a possible source of error generation, not as an error detector. However, there are tasks that can be formalized for the keypuncher which will help in the detection of errors, without reducing the productivity of the keypunch operation. In fact, the keypunch location is an excellent internal audit location, for here estimates of errors generated by the recorder, missed by a data checker, and keypunched, can be determined. Also, errors generated by the keypuncher can be estimated by comparing source forms with punched cards.

Local Computing

Local computing is separated from the keypunch location only in generic terms. There are many cases where the computing facility also houses the keypunch facility and could, in fact, be outside the door of the computing room. The physical location is not important. What is important is the power of the computing available at the facility and the amount of local reports that is generated using data that have passed from data generator, data checker and keypuncher. The detection procedures at such a facility would require formalization, and they are considered part of a formal internal audit.

The local computing facility is the first level in the information system where machine readable data is the basic input. At this location, the detection techniques will be mainly computer-assisted as opposed to manual or mechanical aided detection techniques. Also, the local computing facility is the first level where management reports can be generated. The requirement for local management reports may increase the need for error detection procedures beyond that actually required for all the data. This means that selected data elements may receive more detailed error detection procedures than other data elements or all data would receive the same degree of error detection. If the latter is true -- that all data receive the same degree of detection -- then higher levels of the information system may not be required for perform such a detailed error detection scheme.

Whether this uniformity of detection is good or bad depends on the system accuracy specifications for data elements at the higher levels of the information system. Such specifications could be quite different from the accuracy specifications at a lower level such as the local computing facility.

Central Computing

The use of central computing refers to that location or locations where the data are used for the highest level of management information. The data can also be aggregated for use by lower levels, but the main requirement is that there is no higher computing facility that receives data from a local computing facility.

The error detection function of the central computing facility is completely associated with computer-aided programs. It is all inclusive in the degree of detection performed. The facility must provide all the techniques required by the system specifications for accuracy. As stated earlier under local computing, the degree to which this detection is performed is a function of the amount of detection that has preceded this level. For some data elements, there may have been no formal detection procedures; and a great deal of detection may be required at the central computing location.

Data System Analysts

System analysts function in the error detection process differently from the other detection locations previously discussed. The analysts are looking at the information output of the system rather than the data input. In this capacity, the analysts function as data monitors rather than data checkers. Their responsibilities are to detect errors in the output relative to the input.

It is sometimes more obvious to see data errors in output than to detect such errors individually from the input. The one assumption is that the output data have not been aggregated to a degree that makes translation back to the input data impossible.

Some examples of the error detection would be relationships between data elements that can be easily seen from a formatted output, but practically impossible when observed individually. One such case could be identical part numbers (same part) but a wide range in price, such as could happen by a digit inversion.

While the functions of the data system analysts are different from the other detection location functions, in the sense of output as opposed to input, formal error detection procedures are still possible. Again, the degree to which the location is used depends on the accuracy specifications of the system.

Information User

As discussed in an earlier chapter, the system user has certain fears about the information presented to him. It was also stated that the system designer should make certain options or services available to the user. If such services are available, the user will, in turn, provide services for the error detection function. The actual method for providing the services will vary with information users. It will be either in the form of a request for some additional service, or actual communications with the information system manager. The communications will take the form of describing an exception that has caused the user to question the correctness of a specific report or portion of a report. Such requests and communications should be analyzed as to their cause, for this is a permissive form of error detection.

There could also be formal error detection procedures established at this location. Such formal procedures would constitute a portion of the information feedback loop to the system managers and should be encouraged.

Criteria for Selecting Detection Procedures at the Locations

In the previous section, seven detection locations were described. As mentioned before, not all information systems will have or require error detection procedures at all of the seven locations. However, if the assumption is made that all locations will have a positive and formal role in error detection, considerations must be given to the question of the depth and range of detection at each location.

Considerations concerning depth and range are necessary in view of two conditions inherent in the design of new information systems. The first is associated with the actual data required by the system users, and the second is the changing specifications for data accuracy at the different organizational levels of the system. In the first case, some new data elements will enter while others will leave the information system because of imprecise evaluation of user requirements at the outset of system design. While it is hoped that such changes will be small and have little effect on current procedures, such is not always the case. To minimize the effect of such changes, more detection locations should be considered than might be necessary if the exact user requirements were known.

In the second case concerning data accuracy, changes to specifications at different levels can increase or decrease the degree of error detection. Again, to minimize the effect of such changes, an overstatement of detection locations is better for the short run.

In selecting the depth and range of error detection at the different locations, the above mentioned uncertainties must be kept in mind. Statistics and location characteristics can be developed that will be beneficial in the development of actual error detection procedures concerning these uncertainties.

Expected Workload

The expected workload at the transfer point is intended to describe the amount of data entering the system at that point. At any particular level, there may be a wide range in the expected workload, and different resources would be required to perform the same degree of detection required by the procedures. This would be a system requirement to insure that the next level of detection could start from the same base.

In order to estimate the expected workload, it is necessary to know the specific data to be collected, the

form of collection, and the time requirements of the data to flow through the system. The most critical of the three parameters is that of time. If the time requirement is very short, then the alternatives available for collecting the data are reduced. The amount of time available for the detection process is also reduced. In addition the amount of the reduction in error detection due to the time requirement is related to the volume of the workload at each of the collection and data transfer points in the system.

Under the assumption that the time requirements are not so short as to require an on-line real-time information system, estimates of the workload (raw data input) can be predicted from the various points of collection. As a first approximation to the workload, the amount of static data that would be available for automatic source data collection should be estimated. The volume of these data would then reduce the amount of error detection required from the source. The remaining volume (total less source data automation) would then be a candidate for detailed error detection.

While the data that were collected by source data automation techniques should not be forgotten, less detailed error detection procedures would be needed for the data that had the benefit of automatic collection techniques. That is, all of the procedures associated with format, omission and data element transposition errors would not be necessary. The error detection needed for the automatic source collection would be integrated with the other error detection procedures dealing with the selection of the proper code.

If the system contains the requirement for machine readable data at the earliest possibility, the mechanical aids necessary to meet the data accuracy requirements of the system must be accurately forecasted. However, meeting the specific requirements demands exact foresight, which is not possible in light of all the uncertainties connected with the final data input elements and their error structure.

In view of such uncertainties, it seems only reasonable to develop an expected workload for each level of the information system. In addition, it is necessary to develop within each level the expected workload of each source or transfer point. From such statistics, a basic error detection procedure plan can be developed for each level. Unique variation to the plan could be applied to specific collection or transfer points depending on the expected workload.

During system design and, in particular, during the development of operational system specification, the system data elements and their source are defined. Included are the actual sources of the data, the method of transmission, the frequency, the volume, the form of the data. The specification should also include the output form of the data in terms of their appearance in specific management reports. The availability of such statistics determine the extent to which each level, and each source within the levels, can perform detection.

The Expected Error Count

The expected error count is an important criterion for the selection of both the range and depth of error detection procedures. When there is a high probability of a data element being in error, there is a greater requirement for providing the means to detect such an error. Also, the more important a data element, i.e., its information value, the greater the depth of detection required to insure its correctness.

When data are obtained from automatic source data collection devices, the expected error count is much smaller than data collected by human observation and recording. Also, the form of the data collected has an important relationship to the error count.

Earlier, Chapter III described some research in coding of alpha and numeric data. The results of such research would be helpful in determining the expected error count for

specific data elements.[1] In addition, basic statistical decision theory can be used to generate probability distributions of error counts based on different assumptions. Such statistical models should be developed and maintained during the initial operation of a system.

The use of such models would be to obtain continuous error statistics on the most troublesome data elements, i.e., those that have the highest error count. Having such data would be feedback to the system manager as part of the internal audit of the system. Decisions could then be made as to what additional action should be taken to assure that all errors are being detected or to find the cause of the large error rate and attempt to correct it, or both.

Key Success Variables

A key success variable is defined as a function, technique, method, or resource that interacts with the three major criteria parameters of error detection. These three parameters are: the range of detection, the depth of detection, and the place or location of detection. The availability of the key success variables increases the probability that the detection procedures will be successful.

The success variables are parts of the information system, yet the system designer may have little control over their use or of their quality. In other cases, specific actions or requirements can be placed on the variables to increase their efficiency in the error detection process.

A partial list of key success variables would include:

> Form design - The design of the source form can have the objective of simplifying recording or easing keypunching, but not generally both.[2]

---

[1] In particular, the results of Chapdelaine on length of digits; the work of Miller and that of Owsowitz and Sweetland tend to provide insight into expected error counts.

[2] E. W. Kimball, Reliability Handbook, ed. by W. Grant Ireson (New York: McGraw-Hill Book Company, 1966), Chap. 9, p. 6.

Development of aids - Certain aids can be developed to enhance the detection process. Examples include short lists of the most commonly used codes, the most common errors committed, etc.; such lists will be unique to different parts of the system, but general guidelines could be developed.

Standard procedures - The system procedures developed for completing the various actions should be as standard as possible. As the number of exceptions increasesfrom a standard method, variation and errors are introduced into the data because of individual interpretations to the exceptions.

Documentation - Associated with standard procedures is the requirement for precise documentation of those procedures. The documentation should be prepared in such a manner that there is one and only one interpretation of its meaning.

Adaptability - The degree of variability or flexibility that can be built into the system in anticipation of changes to the data collected, method of collection, and changing organizational objectives and requirements. One such example is the separation between the collection form, the keypunched cards, and the computerized data handling system.

Micro-input - A concept of collecting all required data in their smallest form. While such collection may increase the volume of data collected, it allows for stricter error detection techniques at the lower detection locations.

Feedback loop - A vital link throughout the system is an effective feedback. Such a feedback loop to generators and detection locations will provide the necessary communication for maintaining and improving procedures in data generation as well as data error detection.

Interest and motivation - Of all the key success variables, interest and motivation are probably the least changeable by the system designer and manager. But one of the most important in selecting the error detection procedures to be used at the different locations. In general, the techniques employed should not require more of

> the individual doing the work than can be beneficial to that individual. What is intended is that the procedures must assume each individual has a place in the system. It must also be assumed that the individual will perform error detection in a manner which will reward the direct interests of the level in which the individual must operate. As an example, the keypunch operator cannot be expected to do elaborate error detection which requires looking up relationships in a catalog. However, if a most common list of codes unique to that location are easily available (i.e., tacked on the wall beside the keypunch machine), the operator would find it beneficial to use the list. The list would be beneficial because it makes the keypunch operator's job easier. Consequently, the results of the job are more correct in the eyes of those who are in authority immediately above the operator. There are other examples of the same kind of motivation that prevail at other detection locations, and these examples should be taken into consideration when deciding on the particular error detection procedures to be used.
>
> Also, the individuals usually hired or detailed to the task of error detection do not understand or in many cases do not care to understand their relationship to the total information system. Because of the uncertainty associated with the actual motivation of individuals in the detection process, the procedures should be as simple and exact as possible.

The above described key success variables are not intended to be all inclusive. They do, however, tend to describe the kinds of internal and external factors that need to be considered when selecting error detection procedures. The amount of influence any particular factor may have on the detection process can vary within any given detection location. Provisions should be made to account for such variation.

Error Detection Procedures

The development of formal error detection procedures follows the requirements of the system design. The

requirements specify the type of system, the data elements to be collected, precision and accuracy of data collected, and data error detection locations available through the configuration of the system. Earlier, the chapter described seven error detection locations. This section will describe the procedures that could be used at these locations in terms of the data classes and criteria developed earlier.

It should be mentioned that the procedures described in the following sections are novel in that most information systems consider only data admissibility tests as the extent of error detection and correction procedures. In describing the procedures for each detection location the assumption is made that each location is independent of the other locations. Under this assumption, some of the procedures will be repeated at different detection locations. Such repetition should not be interpreted that the use of all procedures at all locations are considered necessary, rather the attempt is to be as complete as necessary at each location.

The Data Generator

The basic function of the data generator, with respect to the information system, is the observing and recording of data. The error detection procedures that are available for use, depending on the kind of information system would include: (1) sight verification, (2) check lists, and (3) internal procedures.

(1) Sight verification

Sight verification procedures would require the data generator to scan visually the completed document. The procedure would be directed at two basic error control problems. The first is that of form completion, and the second is gross data element detection. In the area of checking for form completion, the forms could be either single or multi-purpose. If the form is multi-purpose, that is, if it is used to record data over many different kinds of actions, the data generator must be knowledgeable in the proper data elements to include for each kind of action.

Good form design would highlight the specific areas or data elements required in accordance with the kind of action that was being recorded. Such methods as color code, and separate sections are common ways of underscoring different parts of a multi-purpose form.

When visual scan is the technique, the data generator checks that all data elements are recorded. This is done in the simplest of manners, by just looking for blank spaces where data elements should be recorded.

In sight verification, the data generator should verify those data elements that are unique to the particular location. Similarly, those elements for which the generator needs no additional aids other than the knowledge possessed by the data generator can be sight verified.

Under those assumptions, the data generator would verify all judgmental elements, both static and dynamic, as well as all unique factual static elements concerning the particular location. This is possible and proper since the data generator is the only one who can recall the justification for the judgmental entries. Secondarily, it gives the generator a chance to reevaluate the element selected and to change the recorded element if the reevaluation suggests a change. The static data are basic knowledge that the generator would have, and he needs only to verify that they are correct.

An additional benefit of the sight verification procedure is the possibility of reduced errors at the location where the data on the source form are transferred into machine readable form. The benefit is that as the data generator rereads the source form in order to check for completeness and correctness, illegible codes or characters that could not be read or could be misinterpreted, would be made legible.

(2) Check lists

A procedure which is more detailed than simple sight verification is that of a check list. The check list can come in many forms. The principle, however, is the same,

and would work in the following manner. The data generator, upon completing the form, would be asked a standard set of questions. These questions would be provided on a one-page check list that would be available before the data generator submits the completed form to a higher level. The type of questions on a check list would be directed at both the completion and correctness as well as the legibility question. For the unique static elements, the questions would be of the "is the location correct and complete" kind. Here again the check list is relying on the knowledge of the data generator.

The dynamic elements will require more exact statements related to the specific data elements and to relationships between the data elements. Included would be a list of the most common errors computed at the data generator location.

Questions concerning admissibility validation would be possible at this point as part of the relationships between data elements. The statements would be of the kind "if block x is c, d, or e, than block x must be 5, 6, or 8."

It is not necessary to include all such relationships on a check list. The attempt is to cover each data element as to range rather than as to depth. In this manner, the whole form will be checked, and, by association, the data generator will provide some additional depth that was not explicitly considered, but possibly was anticipated by the check list.

(3) Internal procedures

The basic characteristic of such a procedure is that a requirement is placed on the source generator to provide complete and correct data by the internal workings of the system itself. The data generator views this particular error detection procedure as an integral part of the system rather than as an error procedure he is required to perform.

The detection of errors through internal procedures requires the data generator to obtain some reward for

the completeness and correctness of the form. The reward can generally be something that makes the job easier or quicker. To do this requires that the source form be completed during the action and used as a request or purchase order for acquiring something needed for the action. Examples would include authorization to draw material, commissions that are payable, wages tied to piece work, speed of response, and recognition by supervisors. In fact, if the data generator does not follow these internal procedures, the completion of his job is much more difficult, if not impossible.

The Data Checker

The basic function of the data checker is to validate the data prepared by the data generator. The procedures open to the checker are similar to those of the data generator. There are, however, additional procedures which the data checker can use if the checker is removed from the source generating location.

In order to discuss the error detection procedures available to a data checker, two different checker locations will be assumed. The first is located near or with the data generator; the second is located away from the data generator. In the first case, the data checker is assumed to have additional duties and the duty of error detection is somewhat secondary.

In the second case, where the data checker is removed from the data generator, it is assumed that the checker's primary function is error detection. Such a situation could come about where many data generators submit their completed forms to a central office within the data generation locations. Examples could include: (1) Local offices of the Internal Revenue Service which process all returns submitted; (2) Branch offices of an insurance company that receive all of the premiums due on policies of that area; (3) A central office aboard an aircraft carrier that receives all maintenance documents before they are submitted to higher levels;

and (4) An office in a job-production shop that screens production data forms before they are passed to a central processing or to a local processing activity.

While the two different checkers have the same job, the motivation is different. For the data checker located near the data generator, the checker has an operational function as his primary duty, while in the second case, the primary function is detecting errors. Since the second checker is getting paid to detect errors, he is more highly motivated than the checker who is doing the detection as a secondary function.

(1) Types of error detection procedures in the vicinity of data generator[1]

(a) Debriefer

The data checker is a co-worker who must sign-off on the completed form before it is accepted at the next level. The procedures open to this data checker are the same as described under (1) and (2) under the data generator, i.e., the visual verification and the check-off list.

However, there is an additional procedure available to this data checker and it can be classified by the term debriefer. It is possible for the data checker to establish a direct communication between himself and the data generator. In many cases, the data checker, as a co-worker, is as knowledgeable about the action being reported as the data generator. Since this is generally the case, the data checker will have knowledge about the unique static elements associated with the action. Furthermore, through the scheme of the check list and direct communication, the co-worker can question the data generator's selection of the judgmental elements that were recorded. The use of this procedure will not only check for form completeness and correctness, but will be an additional check on the judgment of the data generator.

---

[1]See page 54 for a description of data checkers available at this location.

The judgmental checks, as performed by the co-worker, would be a technical dialogue between two qualified workers, both of whom have similar expertise on the object of the action. Because of this technical dialogue, those data elements that had the benefit of two opinions or judgments instead of one should be more accurate.

This does not mean that all judgmental data elements will have the benefit of two opinions. The workload may be so demanding that not all data elements and documents can be checked. In such cases, data element priority procedures would dictate the elements to be checked, while sampling procedures would dictate the number of documents that should be checked. Either the priority procedures or the sampling procedures or both could be used depending on the amount of time available for data checking at the location.

(b) Cross-reference lists

If the data generator is required to work alone, as in many systems, the system has lost the benefit of two opinions. There are, however, procedures available at the supervisory level that will help to compensate for the loss of the co-worker concept.

In general, the supervisor has under his control the work plan of each of his workers. This means that the work each is to do is planned for some period in the future. The time period may be as short as an hour or as long as a week depending on the environment in which the system is working.[1] Because of the advanced work planning, the supervisor has knowledge of the action being reported by each of his workers. Also, the supervisor has some degree of expertise or technical knowledge available that can be applied to the judgmental and non-unique factual static data being

---

[1] In on-line real-time systems this step is passed, since all error detection is performed either by the data generator or by computer-assisted programs, unless the computer is the supervisor. Then such a procedure is quite appropriate.

reported. With this basic knowledge, the supervisor is in an excellent position to perform checks on report completion and correctness. This is possible for the unique static data, the non-unique static data associated with the object of the action, and the judgmental data.

In addition to the procedures available at the data generator's level and the use of the debriefing technique, the supervisor has planning sheets and short cross-reference lists available for checking the reported action. The kind of cross-reference lists that should be considered are those that are directly associated with the major area of the supervisor. That is, a supervisor will generally be responsible for a much smaller set of functions than are included in the complete system.

Since each supervisor is unique in terms of his responsibilities, the cross-reference lists are unique, but their aggregation is equal to all the functions that are to be reported by the system. Tailoring of the cross-reference lists can be described as the separation of the total functions into select sub-sets. If the supervisor functions are standard over all source data locations, the tailoring of the cross-reference lists is simplified. It is, therefore, well understood by all levels that interface with the supervisors level.

In addition, if data are collected at more than one supervisory level, the supervisors at the different levels will have different check lists tailored to their responsibilities. Also, not all documents need to be checked; sampling procedures could be used to minimize the amount of supervisory time involved in checking the documents.

Examples of the tailored lists would be the list of equipment and machines, job numbers, and parts most commonly used for a supervisor in a job-shop or maintenance position. A supervisor in an inventory environment would have a list of the stock numbers, their unit price, their location, and knowledge about their application to different equipment.

(2) Types or error detection procedures for data checkers removed from the data generator

There are information systems wherein the operations of the information system do not follow the line authority of the organization, such as production planning and customer billing. In such instances, the line authority has no interface with the reporting system and cannot be used in the error detection process. Where such operations exist, there is a capability to have data checkers within the data flow process.

(a) Check lists and cross-reference lists

When the data checker is placed in the vicinity of the data generator, but removed from the line authority, the error detection procedures are similar to those of the co-worker and supervisor. In particular, the data checker will use visual scan and check lists. There are, however, modifications to these two procedures, depending on the experience of the data checkers. In most systems, the data checkers are not to be considered experts or even knowledgeable on the technical aspects of the recorded data. The checkers are more like auditors who have the ability to detect discrepancies. With this as the basic assumption, the modifications to the visual scan and check list are extensive.

The data checker in such a position would not necessarily have knowledge of the static or factual judgmental data or of the environmental data associated with the action. To combat this lack of basic knowledge, more detailed cross-reference lists will be required than with the co-worker or immediate supervisor. As with the supervisor, the cross-reference lists should be as concise as possible, but since the primary function of these data checkers is data validation, the range and depth of the lists can be increased. The primary check lists should inform the data checker of functions or jobs that are unique to the particular data generator location.

The lists would include data that range from the very basic static data such as organizational location through the more complex such as material identification codes. In the case of the identification codes, the depth could include particular equipment located at the various data generator locations.

In addition to the primary lists which are closely tailored to the various data generator locations, secondary lists of cross-reference or joint relationships of the most common data element codes would be available. The depth and range of the relationship lists are a function of the workload, the initial range of unique data element codes used, and the value or priority of detection assigned to the checkers.

(b) Bounds and limits

The data checker has a simple procedure available for use on some environmental and judgmental data. While it is not as exact as that available to the co-worker or supervisor, it will increase the error detection ability at this location.

The concept involves putting logical bounds and limits on selected elements. Such a concept would be used for the dynamic data elements of calendar dates, clock hours, meter readings, distances, quantity of parts used, man-hours and other dynamic elements. The data checker would be provided with specific procedures for evaluating the recorded data.

In the area of bounds, the bounds would be set by the logical context of the data elements. An example of such bounds would include the date. The broadest bounds would suggest that the date entry must be a legal date. A second limit within that bound would be to place an upper and lower limit on the actual date within the legal dates. For instance, if the information is reporting historical facts, the upper limit cannot be a date that is in the future. In fact, it can be no more current than the date the checker reviews the document.

The lower limit or past date in such a historical system would be set by the means of communication, and the function or operation being reported. For example, the Navy's Maintenance and Material Management Information System has a data lag of six to eight months before all actions generated in a given month are received by the central computing facility.[1] For this information system it would be necessary to accept a date that is eight months old as a limit on that data element.

However, for the data checker who is reasonably close to the source of data generation, a much closer lower bound could be placed on the data element. This new lower bound would be established according to the method of communicating the document to the data checker plus some variation to account for any randomness in the communication method.

The techniques for placing limits on the data elements will generally be statistical in nature. The selection of the statistical procedures at this level must take into consideration the system objectives in relation to the end use of the date, as well as the key success variables associated with this level.

If the assumption is made that the most detailed error detection possible should be performed, then the objective is to challenge all recorded data elements that seem to be in error. It is most important to remember that the object is error detection. Checkers should not be permitted to change a data element if they think it is in error. The procedures should be very specific as to what is considered an error for detection purposes.

One method of insuring the proper limits is to develop the limits at a higher level and transmit them to the data checker location. In this manner consistency between the data checker locations with standard limits would

---

[1] S. W. Timmerman, "Timeliness of Ship Data Submission to Maintenance Support Office," Technical Report 2029-120126(S), (5 May 1967).

provide additional control over the data checkers as well as the detection process.

(c) Consistency checks

The two classes of procedures described for the data checker can be characterized as one for static data, using cross-references and check lists and one for dynamic data utilizing bounds and limits. An additional procedure that is available to this level is that of consistency over and between data generators. While the other procedures were associated with the detail of the individual reports, this procedure is directed toward the detection of errors in a more aggregated context. For example, the procedures would use such aggregates as the total documents submitted, variations in materials consumed for like jobs, and demands placed on an inventory system by different users.

The means by which these procedures would be developed would include such techniques as recording cumulative totals of a specific data element over a specific time period, developing ratios between selected data elements, and the use of quality control charts.

The data checker that is removed from the data generator location does not have the ability to communicate directly with the data generator. Since the data checker cannot use the debriefing procedure, and since he is not as technically oriented as the co-worker, the procedures available are more detailed. The procedures take more time to perform than like procedures located at the data generator's location.

The best possible situation would be for the data checkers to be located near the data generators -- close enough so that the checker could have direct access to the data generator. Such a relationship would accomplish two things: (1) the data checker could act more as an interrogator challenging the recorded data entries; and (2) the close proximity would allow the checker to recapture the correct data before the generator forgets the action. The amount of such detection and interrogation depends on the

workload at the location. Sampling procedures could be used in determining the documents to be checked. Data element priority lists could be developed for the specific elements of the document that should be checked.

Both the sampling procedures as well as the priority lists could be based on past history of errors committed, both for specific data elements and system errors to determine the actual sampling plan. A successful sampling plan would minimize the number of data checkers necessary, and still insure a high degree of data accuracy from the data generators.

Keypunch Location

The keypunch location will be divided into four separate sections or divisions. These include an incoming data checking section, the actual keypunching section, a verification, and finally an outgoing data checking section. In addition, the keypunch and verification will be described as one section when optical scanners are used.

(1) Incoming data checker section

This section has three functions in the error detection process. The first function is that of sorting the incoming reports into homogeneous groups or keypunch program packages according to the card formats that are used by the keypunch operation.

The second function involves report completeness and legibility. The performance of this function will result in reduced errors generated by the keypuncher due to the inability to interpret the characters on the report. A visual scan of the report for completeness will identify errors of omission that would be produced by the keypuncher. The detection of errors associated with completeness can be performed by visual scan or by the use of a template, where the data checker has limited knowledge of the different formats that are available.

The third function is the detection of data element errors. This function is similar to those already

described for other data checkers, and the procedures for use by the data checker at the keypunch location would be the same. The degree of the use of these procedures would complement those used at the lower levels.

(2) The keypunch section

The keypunch section is composed of keypunch machines and keypunch operators. The primary function of the section is transferring the reports into a machine readable form. The error detection procedures available to the keypunch operators are limited to the use of program control cards, to individual familiarity with the system, and to visual scan. The visual scan offers the most potential source of error detection for the keypunch operator.

At this point it should be pointed out that while the keypuncher is performing system error detection, i.e., finding errors others have made, the actual function, if viewed from the keypunch operation, is error prevention. That is, the keypunch operator has not yet committed the data to machine readable form and he is trying to prevent errors.

The techniques that would be helpful to the operator are those that increase the ease in which the job is accomplished. The use of batching the reports which contain similar data is one such technique. Under program control some of the information can be duplicated from the previous punched card without operator interjection. This speeds up the keypunching rate, eliminates the possibility of errors for that section of data, and makes the job easier. Error possibilities are reduced when the flow of repetitive data is interrupted by the placement of a non-standard form in the sequence. Such a non-standard form will alert the operator that the next set of reports contain different repetitious data which must be keyed into the first punched card.

Error detection performed by the operator relates to basic procedures associated with the punching of the reports. If the keypunch section is large enough to support

specific machines and operators to selected sub-sets of the different report formats, small cross-reference tables can be provided to the operators for their use. The data provided on the cross-reference tables are specific to the subset of tasks or actions being punched at a particular machine. The table will be small and should include the top ten or 20 data element codes most used, or most frequently in error.

Additional formal instructions or procedures should be instituted concerning the omission of data. Reports that are received, with omissions by the keypunch operator should be set aside for evaluation by the data checker; the same is true for illegible data elements that are not easily recognizable.

It should be remembered that the keypuncher has little motivation or ability to perform error detection procedures, unless the benefit gained is greater than the time and effort expended. As a result, all error detection performed by the operator should be considered as a gift. The procedures that are most helpful to the operator are those that require minimum time to complete from the operators point of view. This means that procedures such as the visual scan, programmed control cards and quick "look up lists" are most important.

(3) Verification section

The verification section is composed of verification machines and operators. The primary function of the section is the review, by direct duplication, keystroke by keystroke, the reports punched on machine readable form. As a review technique, the verification section has the same input records and the same program control cards as the keypunch section. The necessity for a verification section, or for the verification function at all is questionnable. In Chapter III, studies were referenced where the verification detects very few errors committed by the keypuncher. It is quite possible that this function could be eliminated and the resources better used. For example, the Bureau of the

Census eliminated verification of the 1960 census data. Statistical studies indicated that the large sample size would more than compensate for the cost and effort expended by verification. One suggestion is to use the resources in the output data checkers section where the keypunch operation could be sampled, without full verification.[1]

However, if this section is used, then the error detection procedures available are the same as those of the keypuncher, but the function of the verifier may interfere with the error detection process. As an example, the verifier is to duplicate, keystroke by keystroke, the data punched by the keypunch operator. If the keypunch operator has punched the code as recorded on the report and the code is a member of the code set, but not the correct code for the action, there is little chance the verifier will detect an error. This is because the main function of the verifier is duplicating the data recorded on the document.

Currently, the only use of the verifier is to detect errors made by the keypunch operator, not errors made by the data generator. When the importance of the current function is analyzed, it may be beneficial to include more functions into the verifier's job than is now considered desirable. The additional functions would include the use of key word lists, and standard unique cross-references.

(4) Optical scanning section

In information systems where the data are transferred to machine readable form by a character recognition service, the functions performed by the keypuncher and verifier are incorporated in the device.

Current character recognition devices consist of two kinds: Magnetic Ink Character Recognition (MICR) and Optical Character Recognition (OCR). While MICR is used primarily for processing bank checks, the technique is available for other applications. The current technique will, however,

---

[1]Herman H. Fasteau, J. Jack Ingram and George Minton, "Control of Quality of Coding in the 1960 Census," Journal of the American Statistical Association, (March, 1964), pp. 120-132.

read only numeric information, and all the machines are tied to a standard font size (E13B). The MICR machines are all equipped with character error detection devices, which, while quite accurate, will read from only a small portion of the document, as specified by the American Bankers Association.[1]

The OCR devices will read alpha-numeric information of a constant font, and several will read different fonts so long as the machine can determine which font is being used.

While the current OCR equipment is capable of reading pages of typing, or portions of pages controlled through a pegboard or a computer, the errors introduced by normal office functions are still bothersome. For maintaining minimum scanning errors, the user of OCR equipment must have control over carbon copies (blurred characters), typewriter ribbons, key pressure, paper surface, as well as dirt smudges, ink smears and other handling hazards.

The use of OCR devices will require a different kind of error detection procedure than discussed previously. Because of the limited applications of such devices, it is sufficient to state that the use of OCR devices has not progressed to a formal state-of-the-art where they are useful to the general information systems designer. There are, however, applications of OCR for documentation retrieval systems, there the input data are prepared in clear text and on clean typed pages. Its use in such systems is of marginal interest to this paper and further references will be limited to discussions concerning MICR techniques.

(5) Output data checkers

This section of the keypunch location has three functions to perform in the error detection process. The first task of the data checker is that of internal audit, the second is data consistency and the third is data accuracy.

---

[1]U. S. Government, General Services Administration, Source Data Automation, FPMR 11.5 (Washington, D.C.: U. S. Government Printing Office, 1965), p. 27.

The purpose of internal audit is to obtain information from the system operation about the error detection process. This function is performed by auditing the output cards of the keypunch and verification sections against the original reports that were submitted. The errors detected through this function will be used as additional information to improve the procedures used at the keypunch locations.

The data collected for this function would include: number of errors by kind, such as omission, transposition, etc.; number of errors by data element; errors by alpha or numeric configuration, such as alpha-numeric-alpha versus numeric-alpha-numeric; and combinations of number of errors by data element length. Such error statistics would be by keypunch location and could be accomplished on a sampling basis during the course of system life. The amount of sampling would depend on the data accuracy required by the system and the accuracy which was being received.

The second task of the data checker is to check for data consistency. He must monitor the logical relationships between data sources and data elements. The relationships would check the unique static data elements against check lists such as data source name and code. Other relationships would include factual data elements such as available equipment against recorded equipment, reported addresses against recorded addresses, etc.

The procedures that are available for use would include check lists, cross-reference tables and statistical inference techniques. The check lists would be similar to those used by the keypunch operators, but would have greater depth. This means that the output data checkers will be able to detect errors more closely related to those that should have been detected at the lower levels such as the data generator and data checker location.

The check lists would be organized as any admissibility list including specific joint code set relations as well as numeric or alpha arrangements. In addition, the

consistency checks would (1) detect aggregation errors, such as total number input reports equal to the appropriate number of output records, (2) check that the data have been received from all the required data sources, and (3) check that the quantity meets a recognized standard.

The cross-reference tables would also be more detailed than those available at the lower detection levels. Here the tables would consider a high level equipment aggregation or configuration. An example would be detecting the fact that a naval destroyer performed maintenance on large 16 inch guns which are only found on battleships.

Similarly, these procedures would be appropriate to the data accuracy function as well as the data consistency function. The amount of detail in such cross-reference lists will be in proportion to the workload, machine aids that are available and accuracy requirements for the location.

The data accuracy function includes the use of statistical aids for the determination of standards to be applied against the data classes. Probably the most useful statistical aid would be control charts using the techniques of quality control procedures.

Local Computing

The basic purpose of the local computing facility is to prepare the data for management use. The function can range from a data collection and transmittal task (in a centralized information system) to the task of aggregating, formatting and generating reports for local management use.

The system environment of the local computing facility controls the extent to which error detection is performed. In a highly centralized system, the error detection may be limited to admissibility and relationship checks. In a decentralized system, the error detection may include all of the techniques that are available for any location and would, in fact, be the apex of the error detection process.

The local computing facility is the first and lowest level in the system that receives, as input, the data in

machine readable form. As a result, all of the error detection procedures are either completely or partially computer aided.

(1) Admissibility checks

One of the first computer-aid programs that should be developed for error detection is that of data admissibility. This procedure can be programmed to detect gross errors in data elements that result from alphas recorded as numerics, and vice versa, left and right justification of data fields, simple relationships between data entries, and acceptable dates and time numeric characters.

In detecting such errors, the system would mark the record and data field as an admissibility error, so that correction procedures could be applied to the data at a later time.

(2) Data record counts

A second important computer-aided program is that of record counts. This procedure will detect gross errors in the submission of records. As data are submitted from the different sources through the keypunch locations, a count is made of the number of records generated by the source location. The counts are used to detect sources that are delinquent in their reporting. The lack of data could bias the information presented to management if the sources that were delinquent accounted for a significant portion of a particular material, equipment, dollars or other resources of interest to management.

The data count procedure has other benefits at a lesser level of aggregation. These are counts associated with the number and kind of errors that were detected, as well as the source that committed the error. Such data count statistics would be maintained and used for modification, feedback and training to source levels that were becoming lax or misinterpreting the data recording instructions.

(3) Cross-reference tables and files

A more complex error detection procedure is that of cross-reference tables and files. This procedure will

detect both gross and detail errors, depending on the structure of the files.

Before describing the procedures available with cross-reference tables and files, two points will be clarified. First, the use of large files for detection and correction of data input errors is almost non-existent within the Defense Department, and most of the files needed to do detailed error detection or correction are not in machine readable form or, in some cases, are non-existent. As a prime example, the Navy has no file which relates equipment to a specific ship or a file which relates equipment to a specific aircraft. The Air Force and Army are in the same situation. In fact the Navy Department has just reissued an instruction covering standard data element coding -- a first step in configuration.[1] Such a configuration file would maintain the latest installed equipment by manufacture, federal stock number and serial number to the specific ship or aircraft. The organization and building of such a file requires several years to develop. In the interim, there are several files which can be used to perform at least part of the service.

The second point is to place the description for the procedures using these files at the local computing facility location. The primary purpose in such a decision is to give the local computing facility the benefit of the computing capacity to perform such detection. In addition, if management reports were made at this level, the decision makers would expect the data to meet their accuracy requirements. In many cases these accuracy requirements could not be met without the use of the files or at least the concept of such files. The third reason for placing the description at this location is to describe some problems that would be

---

[1] U. S. Department of Navy, Data Elements and Data Codes Standardization Procedures, SECNAV INST S200.19 of 9 December 1968.

encountered if the function was completely transferred to the local computing facility.

Cross-reference tables and files are defined as those lists of system related data that provide relationships between various data elements contained in an information system. The magnitude of such tables and files depends on the size of the information system. For example, the Navy's stock record file, which contains all the stock numbers of Navy material within the Department of Defense supply system, contains 1.5 million items, with approximately 40,000 transactions to the file each month.[1]

The cross-reference tables and files should be organized in such a manner that they take advantage of detecting the greatest number of errors with as little computer effort as possible. This means that for data code sets that are tightly packed, the exception to the code set should be made available for comparison and the detection process would determine an error if a match occurred.

For example, consider a file that was numeric in nature, containing 90,000 items packed into a possible code set of 100,000 positions. In such a case the file should contain the 10,000 numeric codes that are not part of the code set, rather than the 90,000 that are members of the code set. This assumes that the data element has been checked for non-numerics and none were found. That is, this procedure would be employed after an admissibility check had been performed.

For code sets that were not tightly packed and for joint code set relationships, the cross-reference files would be much longer. Again the computer processing would depend on the joint code set relationships that possessed the greatest expected detection capability for the least computer time.

---

[1]F. Townsend, private interviews held during visits to Maintenance Support Office, Mechanicsburg, Pa., Spring and Summer 1968.

In a tape oriented computer system, a major criterion for determining the optimal number of relationships in a single file is the expected length of the file. Computer time is measured by the tape speed and a major file may require many reels of tape to handle all the relationships. It may be better to maintain several smaller files with shorter relationships and less total running time than a composite file which is contained on many reels of tape.

If such files are maintained in random access storage devices, then other file organization may be more optimal. The main question of concern is to understand that such files are useful in the detection process -- their organization being a function of the operating hardware system and the utilization of the file in performing the detection process.

Such procedures are possible for data belonging to the static and dynamic factual data classes only, i.e., the data elements must have a definable and finite code set. The various procedures depend on the relationships established in the cross-reference files.

The availability of cross-reference files depends on the relationships that are available between the data elements collected as well as other system relationships. In general the cross-reference files allow the system designer to eliminate the collection of redundant data in the collection system.

A major criterion for evaluating the cost-effectiveness of developing cross-reference files is the cost of collecting redundant data. For example, there is no reason to collect the price of an item if the price is available in a cross-reference file which can be organized around a reported data element such as the stock number of the item. In this case the recorded data element (stock number) is static while the price is dynamic.

There are other examples where both elements are static, such as a bank number and name of the bank, or your own name and your bank account number. In both of these

cases, the Federal Reserve System does not record either your name or the bank's name.

As an example of the cross-reference procedure consider the following example which is taken from the Navy environment. Such files and procedures are directly related to other Department of Defense logistic information systems, none of which now uses such a procedure for accuracy checks on data input.

Assume the following cross-reference files are available:

(a) Master file of stock numbers -- This file contains, among other items, the manufacturer, the unit of issue or number per package, the price per package and the manufacturer's part number.

(b) The stock number addendum file -- This file contains the list of stock numbers that have been replaced and what stock numbers replaced them. If item A is replaced by item B and B by C, the list shows A replaced by C and B replaced by C.

(c) The stock number of Component Identification Number (CID) file -- This file lists all the stock numbers that are identified to major components that contain such a CID number. It is a file of all the stock numbers or parts that make up a component.

(d) The Component Identification Number (CID) to Equipment Identification Number (EIC) -- This file lists all the CIDs to EICs. The EIC is a function-location index while the CID is an engineering design code.

The data element to be interrogated is a stock number. The data element has been recorded and has passed the following data detection procedures.

(1) The data generator scanned the report and checked the data element for completeness and legibility.

(2) The data checker scanned the report, saw that it was completed, and that all digits match a simple admissibility check.

It was all numeric and in the proper form. In addition the stock number was not on a short-list of most used stock numbers.

(3) At the keypunch location, the data element passed all of the checks and the keypunch operator punched the data element exactly as it was recorded, as did the verifier. The stock number was not on any list maintained by the keypunch location.

(4) The stock number is now in machine readable form and has arrived at the local computing facility. The stock number has been subjected to a basic computer admissibility check and found to be acceptable, which means that the number is structured as a legitimate stock number. The question is, is it a legitimate stock number? The first check is to see if it is in the master file of stock records. Assuming that the stock record is there, additional information such as price, and manufacturer, and manufacturer's part number can be selected if these elements are needed for other relationships or are additional data elements needed by the system, but are not collected.

(5) At this point detection of the stock number could stop and as far as single detection relations are concerned the stock number is correct. However, if the system accuracy requires that the stock number be accurate in joint relationships, then additional detection is needed. Continuation of such relationships depends on the end use of the data. If engineering analysis is a major part of the use of the data, then it is important to know that the stock number belonged to a particular equipment which was used in a specific application in a particular environment.

(6) Here it is assumed that the recorded CID, EIC and ship number have had independent error analysis performed and no errors have been detected. At this point, the stock number alone does not become the object of the detection, but the complete joint code set relationship. That is, could the stock number

be consumed on this component, which has this equipment configuration and is that equipment located on this specific reported ship? The cross-reference file needed for such a relationship is composed of the basic files maintained by the system. A match of all the independently checked recorded data elements against the cross-reference file will insure that this joint relationship is accurate.

(7) If one or more of the data elements did not match, then those data elements will require additional procedures to determine the exact data element in error. For example, assume that the stock number to CID, and the EIC to ship relationships were correct, but the CID to EIC relationship did not match. To determine which of these two data elements is in error, at least two additional checks will be required. The first would be to see if a match can be obtained between the ship-EIC and stock number. The other check would be between the ship-CID and stock number. If one of these checks did not obtain a match, then the data element involved in the no match is in error. If both of these last two checks obtain a match, then the next match will be from the cross-reference file containing only the CID and EIC.

Before a match is made at the CID-EIC level all attempts to check for additions, deletions and changes to this file should be made to insure that the file is as accurate as possible. Once the file updating has been accomplished and a match is not obtained, then the CID is considered in error.

In determining the number of different relationships that are possible for such cross-reference files, Table II displays the various data elements that could be in error for the above example. Table II is separated into five sections to display both the single relationships and joint relationships that are possible.

Table II

Number of Single and Joint Relationships
Between Stock Number, CID,
EIC, and Ship Number

Ship	EIC	CID	Stock No.	
				All Elements Correct
Ship	EIC	CID	$\overline{\text{Stock No.}}$	One Element in Error
Ship	EIC	$\overline{\text{CID}}$	Stock No.	
Ship	$\overline{\text{EIC}}$	CID	Stock No.	
$\overline{\text{Ship}}$	EIC	CID	Stock No.	
Ship	EIC	$\overline{\text{CID}}$	$\overline{\text{Stock No.}}$	Two Elements in Error
Ship	$\overline{\text{EIC}}$	CID	Stock No.	
$\overline{\text{Ship}}$	EIC	CID	Stock No.	
Ship	$\overline{\text{EIC}}$	CID	$\overline{\text{Stock No.}}$	
$\overline{\text{Ship}}$	EIC	CID	$\overline{\text{Stock No.}}$	
$\overline{\text{Ship}}$	EIC	$\overline{\text{CID}}$	Stock No.	
Ship	$\overline{\text{EIC}}$	CID	$\overline{\text{Stock No.}}$	Three Elements in Error
$\overline{\text{Ship}}$	EIC	CID	Stock No.	
$\overline{\text{Ship}}$	EIC	$\overline{\text{CID}}$	$\overline{\text{Stock No.}}$	
$\overline{\text{Ship}}$	$\overline{\text{EIC}}$	CID	$\overline{\text{Stock No.}}$	
$\overline{\text{Ship}}$	$\overline{\text{EIC}}$	$\overline{\text{CID}}$	$\overline{\text{Stock No.}}$	Four Elements in Error

In general, the number of such combinations is the sum of the binomial coefficients $\binom{n}{x}$, where n is the number of data elements in the relationship and x is the number of data elements that can be in error.

For the example n is 4 while x ranges from 0 through 4, or

$$\sum_{x=0}^{4} \binom{4}{x} = \binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4}$$
$$= 1 + 4 + 6 + 4 + 1$$
$$= 16 \text{ possible error relationships .}$$

However, since each data element was found to be correct independently, the number of possibilities decrease as they do when you consider only the adjoining relationships that are of interest. This reduces the number of three relationships for two data elements in error, two relationships for three data elements in error, and one relationship when all four data elements are in error for a total of six joint relationships that are of interest.

The use of the cross-reference file procedure, while limited to factual data, provides an important extension of error detection capabilities at the computing locations. To recall an earlier statement, the range and depth of the procedure depend to a great extent on the system requirements for accuracy at the local computing level. This is in addition to the cost of providing the various cross-reference files at the many local computing facilities.

If the accuracy requirement for a complete file at the local level cannot be justified, the development of a partial file seems reasonable. A partial file, either in range or depth, would be beneficial to both the local decision process and the central computing facility through the elimination of part of the detection process.

There are, however, several problems associated with decentralized cross-reference files. One of these problems is the file maintenance and updating problem at the

different local computing facilities. Another problem is the synchronization of the procedures at the central computing facility over all of the local computing facilities. Both of these problems are interrelated, and careful planning between the local computing facility is a must for successful operation.

(4) Statistical techniques

The last set of procedures to be discussed for possible use at the local computing facility are those associated with detecting errors through statistical procedures. All of the techniques would fall under the general title of testing for "outliers" or "extreme deviates." The techniques available fall into two general classes of statistics. The first is that of statistical quality control, and the second is that of statistical inference.

The data to which these procedures would apply are the dynamic factual and judgmental data classes. Examples of such data elements include: clock time, calendar time, elapsed time, man-hours, and cost of resources, to name a few. While the specific data elements are dynamic in nature, statistical procedures can be used to develop tests that will detect an error in a specific data element. The procedures will also detect more aggregated system errors associated with completeness or timeliness of the data input function.

(a) Quality control procedures

One of the simpler methods for detecting errors in dynamic data is to use the principles of quality control. In the most simple application, the procedure would require the calculation of the mean (arithmetic average) and the variance of the data element for some period of time. The values computed would then be used to set limits of acceptability on future data collected.

The intent of the quality control procedure is to build limits around the data elements to detect reported entries that seem to be out of norm with the average. In

general such a procedure would check each new observation of that data element to determine if the new observation is within the limits set for that data element. To establish the limits for a dynamic data element it is necessary to:

(1) Define the relationship between the dynamic data element and a static element.

(2) Compute a mean and variance for the dynamic data element for some period of time within the environment of the static element.

(3) Establish the limits on the dynamic element according to the decision rule for the degree of error the system will accept.

Once a limit is chosen, a probability can be associated with that limit. It should be understood that if the limits are too wide, then the probability of detecting an error decreases. Establishing limits that are too narrow increases the chance of calling an observation an error when in fact it is not an error.[1]

As an example of the procedure, consider the following. There is a requirement to establish limits on the amount of direct labor spent on servicing aircraft before flight. The servicing amounts to checking tires for proper air pressure, cleaning windows, refueling the aircraft, and transporting the aircraft to its ready position. This data element is reported as a total man-hour figure for each aircraft serviced. The aircraft are of the same type and kind so that any variation in total time is due to variations in performing the service. Data collected for the past six months have the following characteristics.

There were 10,000 records received; the average time was computed as 4.5 hours. The variance was 1.1 hours.

---

[1]The statistician would refer to these as type I and type II errors. See W. J. Dixon and F. J. Massey, Introduction to Statistical Analysis, 2nd ed. (New York: McGraw-Hill Book Company, 1957), for formal treatment of the subject.

What should be established as the limits for direct labor associated with service time? Under the assumption of large sample size, and a normal distribution, the limits could be set at:

4.5 ± 1.04 to include 68 percent of the expected times;
4.5 ± 2.08 to include 95 percent of the expected times;
4.5 ± 3.12 to include 99 percent of the expected times.

At 68 percent, the limits are 3.46 to 5.54 hours. Such a small range would detect as an error 32 percent of the recorded data that statistically belongs to the same population as the six months data used to create the limits.

At 99 percent, the limits are 1.38 to 7.62 hours; this range would detect as errors only one percent of the records that statistically belong to the same population.

It would, therefor, seem reasonable to accept the 99 percent limits as within acceptable limits, and detect all others above or below these limits. The limits can be revised at a later date if conditions in the system change to a point that makes the current limits unrealistic.

When the above procedure is incorporated in a computer program, decision rules will be needed to update the limits as new information is gained about the system. The revision of the statistics (mean and variances) and the limits can be performed at set intervals of time or as required or directed by the system manager. It is quite possible that revision would be almost continuous at the beginning of system operation, but would become less frequent over time.

In addition, it is also quite possible that the procedure would be used only on a sampling basis as the system became more stable. Another approach to the sampling of data elements would be to use the procedure on the high priority or high error prone data elements.

An additional question that needs to be answered concerns the location that establishes the limits on the procedures. If the data from the local computing facilities are

not representative of all the data element values, but are unique to a select subset, a bias will occur. The bias may cause each local computing facility to have unique limits that would be detecting, as errors, recorded data elements other locations would pass as correct. If all detected errors are not corrected at the location where they were detected, then the central computing facility must have some control over the limits since final correction rests with the central facility.

(b) Statistical inference

The procedures of statistical inference or hypotheses testing are concerned with the evaluation of a group of observations. The statistics are used to determine if the group or a sample of that group is from a population with known parameters.[1]

The application of these known statistical procedures to the detection process allows tests to be made between currently used standards and new data to determine whether the current standards should be updated. The procedures are similar to those described under the quality control procedures except the intent is directed to determining when to change the standards and limits used for detecting individual data elements.

Changes in the system environment or mode of operation may suggest that the criteria for determining errors are no longer valid. The data suggest that the changes have either improved or deteriorated the system. The use of statistical inference procedures would help both in determining the degree of change and in helping to establish the new statistical decision rules.

The simplest of the statistical inference procedures include the general null hypotheses that the new data are from the same population as that currently used for the test standard. Examples of this procedure would include the testing of records received from the different data generators.

---

[1] Ibid, pp. 88-130.

Assume that historical data suggest that each data generator submits X number of records per month. The number X then becomes a standard or mean, and the system is geared to expect that overall average from the sum of the data generators. However, after a period of time it is observed that Y records are being received. To determine if the new level Y is statistically different from X , a statistical hypothesis is described. This hypothesis will test the difference between the currently used mean and a new mean derived from a sample of the data generators. If the two means are statistically different at some pre-chosen probability level, then there is evidence that the current data are different from the previous data on which the standard was developed. At this time a change in the value of the standard should be considered.

This simple procedure can be developed for any dynamic data element which has variation in the code set. It should be pointed out that the variation or statistical variance is an important consideration in the use of this procedure. If the variation among the observations of the data element is large, then the power of the procedure to distinguish changes or to even compute reasonable standards is quite limited.

If it is possible to break the data element into subsets or classifications by a static element, then it may be possible to establish standards for the different classifications. Such classifications could provide the necessary organization of the data element to eliminate the large variance. The elimination of the large variance would then permit meaningful standards to be established for the data element.

The Central Computing Facility

The central computing facility refers to that one activity where the data are retained for use in developing the information used by the highest level of management. It is that one location where all of the data generated by the system come together at one place.

Since the central computing facility receives the data from local computing facilities, all data received are in machine readable form. Error detection procedures used at this location are all computer-aided programs and similar to those employed at the local computing facilities.

The main difference in the use of the error detection procedures at this level and the local computing level is the degree to which the techniques are applied. The range and depth of the procedures at the central computing facility are directly related to (1) the detection that has preceded, and (2) the system requirement for data accuracy.

The central computing facility has final responsibility for the accuracy of the data. In filling this responsibility, the depth and range of the detection procedures previously used must be taken into consideration.

The one location that will have the greatest impact on the depth of the detection procedures at the central computing facility is the local computing facility's capability. The greater the local capability for conducting in-depth detection, the less will be required at the central level. This mainly involves the use of cross-reference files and statistical procedures at the local level.

The concept of cross-reference files is required to insure the accuracy of the joint data element relationships. If these files are not available at the local level, then they should be used at the central facility.

In addition to the cross-reference files, the central facility would have the master file of all the data collected in the system. The master file would allow the central facility to detect errors of consistency. Similarly, more detailed statistical inference relationships between the operation of the local computing facilities could be performed.

A second major feature of the central facility would be the use of ancillary files. These files would contain data received from other systems within the organization and data from outside agencies that would help in the detection

process. As an example, a supply system could provide a file of the parts demanded by user activities to check against the parts reported used by these activities. This kind of detection procedure performed at the central facility would be part of a completeness check on the volume of parts consumed by the system. In general, there are many such uses of the ancillary files for detecting errors associated with the aggregated data element.

It is generally accepted that the higher levels of management are not interested in the details, but the trends, averages, or relative position of the factors within the organization.[1] While this may be true, an information system can have many "masters," with each "master" wanting something different in range and depth. Because of diverse users the detection process must include a wide range of user needs. These needs range from the system engineers who are interested in reliability and maintainability studies, to the top managers who are interested in the trends and relative position information.

The central computing facility is the hub of the information system. All other activities feed data or requests to the hub. In return, the activities receive information about the data they submitted, general system information or information answering their requests. It is necessary for the central facility to coordinate all detection procedures to insure adequate coverage in both range and depth to meet the users' demands for accuracy.

Data System Analysts

One of the roles of the data system analyst is to perform error detection on the information side of the system. That is, the data analyst is to perform error checking procedures to the information contained in the reports. This is to

---

[1]John Deardon, "Can Management Information be Automated," Harvard Business Review, (March-April, 1964).

insure that the proper relationships being used are valid and consistent in the context to which they are being applied. Earlier, in Chapter I, the question of users' fears was discussed. A primary role of the data system analyst is to reduce some of these fears of the system users.

The procedures available for the system analysts include: (1) validation of mathematical computations proposed by the system users, and programmed by the computer facility; (2) detecting data element combination relationships which have no valid relationship; (3) reviewing sample information products and reports in detail with the computer programs and system procedures. The latter is to insure that the data collected are in accordance with what was intended and are reflected by the computer program that developed the report.

As an example of report review, consider the case where a large variation in a particular data aggregation were caused by the method of defining the individual data elements comprising the aggregation. In this case the local procedures for recording the data elements at the various data generator locations were found to be different.

The error detection functions and procedures for the system analyst are different from those described for other detection locations. However, the functions are important for the current as well as the future of the data system reliability.

Information User

The primary role of the information user is his decision making responsibility in the organization. In this capacity the user can provide valuable information about the error detection process.

The scope of error detection can range from increased or decreased accuracy requirements for different data elements, to formal requirements for the initiation of specific error detection procedures. Somewhere in the middle of the scale are the methods of informal detection and feedback to the system manager on which the information user will generally rely.

Each information user is unique in the amount and degree of information received from the system. The amount of error detection each user performs is a function of this uniqueness which in turn depends on the kind of data used in providing the information to the user. Since the information users tend to be informal in their error detection functions, the system designer should formalize the informal communications as much as possible. This means that feedback channels and other formal communication methods are easily available for use by the information users to report on detected errors.

The techniques most useful to the information users are those associated with their own knowledge of the organization. For example, some information users could be technical experts in the organization and operation of the files used and their relationship to the basic data that is being collected. Others may be engineering experts who have basic knowledge concerning the relative value or characteristics of processes that should be reflected by the values in the reports. Others may rely on personnel experience, theory or intuition as methods of detecting errors in the information.

Summary

The chapter developed three major themes. The first was data classes on which data error detection procedures could be applied. The second theme was that of detection locations and criteria for selecting detection procedures. The third theme was the detection procedures to be used at the detection locations and the classes of data.

The data were described by two major classes, static and dynamic. Under each of these classes, two sub-classes were developed -- factual and judgmental. A third class was defined which described the major characteristics of the sub-classes. These characteristics were defined as environmental, descriptive, and action data.

The paper defined seven locations where detection could be performed. The seven locations cover the complete

range from the data generators who observe and record the data to the user who requires the information for his decision making process. The basic criteria for locating error detection procedures at the different locations were discussed under three main topics: (1) the expected workload at any particular location, (2) some expectation of the error count, and (3) a group of key success variables that add to the successful attainment of the procedures at the different locations.

The final section describes error detection procedures to be used at the seven detection locations. The procedures include sight verification, check lists, internal system procedures, the concept of debriefing, error counts, gross reference lists, and statistical techniques of quality control and inference. Each technique has modifications which apply differently to each of the detection locations where the technique can be used. In some cases the range of the technique is modified, while for others the depth has been modified for the location.

The procedures described in the chapter are not detailed for implementation in any particular system. However, they are specific enough in their description and data class characteristics to provide valuable insight into the utilization of the procedures.

CHAPTER V

ERROR CORRECTION PROCEDURES

Introduction

In the last chapter, the discussion centered around the concepts associated with error detection procedures. In this chapter the concepts of error correction will be discussed.

In an earlier chapter correctable and uncorrectable errors were defined and the statement was made that not all detectable errors were correctable. It was also inferred that there may be cases where "the system" would not want to correct a detectable error. In order to determine which errors should be corrected and which errors should pass uncorrected, a concept of error priority is needed.

In addition to the concept of error priority, the location of the error correction procedures within the system will be discussed. The stratification of error correction procedures will be described, but not formalized until Chapter VI.

Error Priority

The concept of error priority can be stated as the difference between the worth of a data element when it is accurate and the worth of the data element when it is in error. This worth is to be considered with respect to the end use of the data element in the information system.

Since the data elements are not unique in their end use, the worth of the data element to the objectives at each management level could be different. With this assumption, the role of error priority has importance at all management

levels. Furthermore, the same error will have a different effect or different priority at different management levels for the same data element.

In order to discuss the error priority, data worth will be described in terms of errors in accuracy. Errors in accuracy are defined as errors which do not reflect the true state of events at the time the recording was made. There can be various degrees of accuracy depending on the characteristics of the data element. For a data element with only two codes, such as yes or no, there are no degrees of accuracy. The complete observation is either 100 percent accurate or 100 percent in error. However, a structural code which has several levels can vary in accuracy as the number of levels increase or decrease. For example, the Post Office ZIP Code is a structured code which can have various degrees of accuracy.

Since information systems have structural codes, and management interest varies with the structure of the code, a given structured code would require different levels of correction. The number of levels depends on how important an error was to the users' decision process at that particular level of the code structure. Hence a need for error priority and data element worth.

Data Worth

As stated earlier, the worth of a data element is associated with the end use of the element at each management level where it is used. Because of the multiple uses of the data elements, it is not possible to place an exact worth on any data element. The best that can be done is to develop criteria for assessing the worth of the data element at the different management levels.

In many cases, the worth of a data element is decided not so much on the whole element as on a particular subset of the codes which make up the data element code set. The subset of codes can, in itself, be dynamic and the content of the subset will change over time. When the dynamics of

the code sets are considered, their data worth, even at the same management level, might also change. Since such factors are the norm rather than the exception, the development of exact measures for data worth have little meaning for error correction procedures. What is reasonable and will be attempted is to describe techniques to be used in determining the worth of data by way of error priority.

There are many factors to consider in determining error priority. The factors include the decision processes involving the data element, uniqueness of data element, the levels where the data are being used, the amount of data already collected, the data class and questions concerning redundancy and alternatives.

(1) Decision process

For each data element collected, the users of the system have described preliminary reports or procedures for using the data. From an analysis of the preliminary reports, it is possible to gain insight into the decision process and computational requirements of the users for the data.

For example, if a particular data element such as cost is recorded to the penny, the users may, through the computations of the report, receive data to the nearest dollar. In some cases, the nearest ten dollars, one-hundred dollars or nearest thousand dollars, may be appropriate depending on the level of management and size of the organization. If such a decision process is used, an error in the decimal part of the cost data element has very little value and hence a low priority for correction.

(2) Data uniqueness

In the area of data element uniqueness, two separate situations are important to error priority. The first is the degree of uniqueness a particular data element code set possesses, and the second is the degree of uniqueness between data element code sets.

In the first case, if the data element code set can be partitioned into independent subsets related to source of generation, two situations are possible. The first is the

situation where the data element is a member of the factual static data class. More important, the data element is a control data element that defines the location of the source of the object of the action. Since the code describes where the action occurred, the error priority is dependent on the degree of attention paid to particular source locations throughout the management levels.

The second case is the situation where the data element code set can be partitioned along lines of the object of the action. In this case, the error priority depends on the degree of detail management that is associated with the object of the action. It is not uncommon for the degree of detail of a data element to vary since codes comprising the data element may have different interest to the system users. Such variation can be stimulated by interest at higher management levels, by outside forces, or by information obtained through the information system reports. Because of the variation that is possible for such data elements, error correction procedures should be developed to cover the highest priority that the elements might obtain.

In the case of uniqueness between data element code sets, the error priority is related to the ability of one data element code set to uniquely define the other data element code. The higher the possibility of a unique match, the lower the data worth of the element that can be predicted. It is not necessarily true that the relationship is reversible. In general, it will not be true except where 100 percent redundancy is provided, since a two-way relationship implies a one-to-one match.

(3) Historical data

In many on-going information systems, data have been collected for several years on the same elements. In particular, the data elements that belong to the factual-static class are used in many information systems for forecasting and estimating standards. While the data elements collected are static in name and function, the individual

codes within each of these data elements are subject to change (dynamic). The new entries within the static data elements force the error priority to those codes or group of codes with the least history.

The requirement for lowering the error priority procedures to individual codes within a data element code set is caused by the lack of history or operational performance on new items that enter the system. Because of the need to monitor or follow the new entry closely, its error priority is much higher than older codes of the same data element. The opposite is true for data which have been collected for some period of time, and on which all of the characteristics are well known.

The requirements for a higher error priority on the new codes that enter the system force the need for error correction procedures for the entire data element to a higher level. This will provide the necessary procedures for those particular codes currently in the data element code set or anticipated for the code set which will be of interest to management.

The complexity of the error priorities at the different management levels allows a subset of a data element or a single data element code to have a higher priority at a lower management level. Such a priority is independent of the priority at the higher management level. When the lower level has a higher priority, the error correction procedure requirements placed on the lower level will be more detailed than those at the higher level. This will reduce the requirement for error correction at the higher level. Due to the variation that will occur in the error priorities over time, the error correction must be available at all levels of the system. This is especially true when it is anticipated that the system will allow additions to established data element code sets.

(4) Redundancy and alternatives

Redundancy in information systems can be defined to be a one-for-one relationship between two data element code sets. That is, the knowledge of one data element automatically defines the code of the other data element.

Alternatives, on the other hand, can be defined as the ability through exogenous factors, not necessarily collected in the information system, that would provide the same data but possibly at a reduced level. That is, there is not a direct one-to-one relationship for all codes.

When complete redundancy is available, one of the data elements has a lower priority for error correction procedures. The decision as to which data element will be the independent element should be based on the ease in which the element can be corrected when found to be in error. While complete redundancy is rare, the redundancy that does occur usually can be found at the lower detection and correction locations in the information system. Redundancy is usually found at the data source generation location in particular.

The method by which the redundancy exists is through some narrative or pre-printed material that appears on the source form as an aid in form completion. As an aid the information will not be processed as the source form proceeds through the system. When such redundancy exists, the error correction procedures are dependent on human observation. Errors not detected at these lower levels will require more elaborate procedures as the data progress through the system. The increased procedures are required since the ability to use this type of redundancy as a method of detection does not exist at the upper levels.

Partial redundancy will exist at different levels throughout the system, and will be associated with subsets of the data element code sets. When partial redundancy exists, the error priority for the subset codes will be lower than the other codes of the data elements. Correction procedures would not be required for the redundant subset throughout the system.

In the case of alternatives, the data element code set can, to some extent, be either estimated or reproduced through the use of exogenous data. The number of different ways available to produce the correct code will determine the error priority of these data elements. If the recorded data element code set is factual, and exogenous alternatives easily applied, it is questionable whether the data element should be collected in the first place. In some cases the alternatives will only be available at the highest locations in the information system, (i.e., computer locations where ancillary files are maintained).

In considering data element alternatives at any level, the amount of extra processing and data maintenance required to produce the necessary relationship must be considered. If the process that produces the relationship is a side product of another requirement, the cost will be small. However, if the relationship is generated to fill a data correction requirement, it may be less costly to include error correction procedures in the formal information system and forget the use of the alternative relationship.

Summary of Error Priority

The error priority assigned to any data element or any subset of the data elements code set is a function of several variables. While it may not be possible to define the exact error priority associated with each data element, techniques for evaluating the priority were described. The error priority is related to the worth of the data. The need for correcting the data is evaluated through a data worth function to gain estimates of the error priority of the data.

Because each data element has multiple uses within the information system, the error priority and the associated data worth will change with each application. In addition, complexity for determining error priority is introduced by the different management levels that use the data. It is reasonable to assume that the error priority for particular

data element code sets will vary over the different management levels. When the priorities are higher at the lower management levels, their error correction procedures will provide data more accurate than required by the higher level.

The Error Correction Procedures

Error correction procedures are of two kinds. Exact procedures correct the data code, keeping the same accuracy intended by the system design, and approximate procedures correct the data code to some statistical confidence that will meet user requirements. Exact procedures are available for factual data, while judgmental data will rely on approximate procedures for error correction.

In Chapter II several conditions were described as necessary for a detected error to be correctable. These conditions are:

(1) That the code contains error correcting digits that enable unique identification.

(2) The code, when connected to other code sets, establishes unique code set combinations that through logical progression are error correcting.

(3) The data element is of such a nature that bounds can be placed on detectable errors.

(4) The coded data element can be returned to the data generator for correction.

(5) The coded data element is of such a nature that statistical techniques can determine the most correct code.

Conditions (1), (2), and (4) are considered exact methods, while (3) and (5) are approximate methods. When the same data element can be corrected by two or more conditions, the exact method should be used if the cost of its performance is not significantly higher than that of the approximate method.

To maintain consistency between the detection procedures and the correction procedures, the latter will be

described in terms of these applicable locations. Since error priority is a function of end use, the procedures will not consider error priority directly.

The Data Generator Location

All data should be recorded as accurately and precisely as required by system procedures. Thus the data generator considers that all data generated at this location have the highest error priority. Correction procedures available to the data generator are limited to the formal system procedures and training included in the system. These formal procedures are more of error prevention than error correction. This is especially true of generators who complete a handwritten form as the method of entering data into the system.

If the source data generator uses a terminal connected to a computer, in an on-line real-time mode, the data generator should be provided with a display panel. Either cathode ray tube or hard copy can be used to review before submitting the data to the computer. Again, this is more error prevention than error correction. It is error correction when both the source document recorder and the terminal user act as data checkers, not as recorders of source data.

Data Checker

In Chapter II five conditions were described for detecting correctable errors. Two of these conditions are appropriate at this detection location. They are: (1) codes have logical progressions that are available for correction, and (2) the data checker is in close contact with the data generator to return the source form for correction or can communicate direct with the recorder. The question of error priority should not be raised at this location. All detectable errors should be corrected. The basic question is still how to correct, not what to correct among the detectable errors.

The procedures used for detecting errors are often helpful in correcting errors. This is true for logical checks where the progression defines a relationship providing a unique code. When such relationships are used at this location, the detection process is the correction process. When the data checker is removed from the data generator, the error correction procedures must rely on relationships, catalogs, check lists, and statistical techniques.

If the data system is large and each of the data elements contains a large number of codes, the data checker should not be asked to provide detail correction for all the detectable errors. He may see that a data element is missing. However, unless the correct entry could be derived from relationships provided internally by the source form, (i.e., a redundant data element) the document should be returned to source for correction.

The techniques for correction by the data checker at this level are limited. The checker must rely on his own knowledge of the system, the check lists, and internal procedures for documenting the recording of an action. The degree of uniqueness that prevails at this level will also be helpful to the data checkers. In many cases the recorded data will contain detectable errors that are correctable because of the uniqueness known to the data checker. One such major area is illegible codes; the data checker may have knowledge of the proper code, or the proper subset of the code, which will minimize the time required to insert the proper code.

Error Statistics

Initially, the data checker will have the same learning curve as the data generator. However, after the system has been operating for some period of time, error statistics will be available to the checker in the correction process. Error statistics are of two kinds: the first kind is statistics compiled by the data checker location and the second is statistics feedback to the data checker from higher locations.

The statistics generated by the data checker location are associated with the kinds of errors made by the data generators in submitting reports. This would include the data elements most often in error, the correct codes for these errors, and a short list of cross-reference data elements which are unique to the data checker location.

The cross-reference lists would contain relationships that are both data elements recorded in the system and information contained on the source form that is not recorded in the system. Such relationships provide redundant data at this level for correction purposes. An example is a name which is not carried forward in the system, and an account number which is carried forward. If the name was preprinted, and the account number was left blank, a data checker can get the correct account number from a cross-reference list of names and account numbers.

The data checker location provides a unique position in the error correction procedure. The uniqueness stems through the ability of the data checker to communicate directly with the data generator. The data checker can police the data generators, encourage better reporting, discover training weakness, uncover procedural gaps, and provide the data generators with a contact for continuity of the total system.

The manager's use of the correction procedure of sending the source form back to the data generators is directly related to the value the managers place on the direct communications between data generators and checkers. If the system managers are only interested in correcting the data, that is, the day-by-day errors, the data checkers will correct all possible errors. This is without informing the data generators of the change or returning the form to the data generator for correction. If the data checkers required the data generators to make all corrections, even those that are easily correctable by the data checker, two immediate results are seen. The data generator would learn

that someone is checking his work and that all errors are considered important. In addition, a type of on the job training would be realized from such a procedure.

When the form is returned to the data generator, the long-run reliability of the input data is improved. The cost of providing for such long-run improvement is the delay in processing day-to-day forms. Such delay can be minimized when the data checkers are in the immediate vicinity of the data generators. However, as the data checkers become removed from the data generators, delays will result.

Possibly, a means for creating the same environment for the data checkers who are removed from the generator could be developed. For example, to correct only those forms containing errors not requiring the data generators' knowledge, such as static or redundant data element correction. All other reports that contain multiple errors where at least one error requires correction by the data generator would be returned to the generator. In this case the generator would be required to correct all the detected errors.

The use of such a dichotomy of work would tend to minimize the number of forms returned, thereby reducing delays in the data transmition. It would also provide for several of the unique characteristics of the data checker location, such as policing, training, and more complete data correction.

Keypunch Location

In Chapter IV the keypunch location was divided into four sections. Such a division was necessary to describe the possible error detection procedures. However, the error correction procedures will be relating to the total location. In cases where the correction technique is limited to a single section, and the section within keypunch is not obvious, the section will be named.

The error correction procedures at the keypunch location are both exact and approximate. The exact procedures

are: short cross-reference lists, look-up lists, admissibility checks, redundancy, and keypunch control programs.

The keypunch location, like the data checker locations, can use the source document in the correction process. At the keypunch location the use of the source form includes all of those at the data checker location plus additional uses that aid the keypunch process itself.

As the documents arrive from the data checker locations, there is a requirement for internal system procedures to effect the transfer of the documents. The requirements should, at a minimum, identify the data checker and give some statistics on the number and kind of source documents that were transferred. This transmittal letter, along with the source documents, will allow the keypunch data checkers to correct the factual data that are unique to either the data generator or data checker locations.

During the process of preparing the data for keypunch, the data have been screened and placed in packages suitable for keypunch. Each package is unique in the data required, and detected factual errors can be corrected by matching the source document with the data checker's transmittal letter. Unique relationship lists concerning the data generators who submitted the data would also be used in making these corrections. In addition, all of the standard batching techniques for data control would be in effect.[1]

Some errors detected by the keypunch location cannot be corrected there. For such errors the alternatives are to send the source document back to the data generator, or to approximate the correct entry.

Taking the two alternatives in reverse order, the keypunch location can approximate the correct entry for only some of the detected errors. The data class most likely is the action-factual data class which contains such data

---

[1]H. N. Laden and T. R. Gildersleeve, System Design for Computer Applications, 2nd ed. (John Wiley and Sons, 1967), Chapter II.

elements as cost of material, man-hours consumed, etc. This is possible if the keypunch location maintains a set of standard jobs and a cost catalog. There are, however, other data elements of this class which cannot be approximated. Such data elements as reason for action or date of action would fall into the first category of alternative, and should be returned to the data generator for correction.

Error priority is introduced in the decision in determining whether the source document should be returned to the data generator. If the data element has a high error priority, the data element should be marked as in error, not corrected but returned to a lower level for attention.

The keypunch location is the lowest location at which decisions concerning error priority should be considered. A main reason for such a decision is the geographic location of the keypunch facility in relation to the data generators. Close proximity allows for quick turnaround of source documents, which affects the data generator's capability to recall the specific action and make the necessary corrections.

The opposite is true as the keypunch location becomes more remote to the data generator. That is, the delay in the receipt of data at the keypunch location plus the processing and transmition time back to the generator could be longer than the recall capability of the data generator.

In considering what error priorities to assign at this level, consideration should be given to those data elements that could not be corrected at higher levels by computer-aided programs. Such data elements would be appropriate candidates for returning to the data generator. However, the use of the data before they reach such a correction location must be considered. If the data are to be used at a management level lower than the correction location, the document must be returned to the data generator for correction.

As the source documents flow from the receiving section to the keypunch section, all detectable errors have been resolved. The errors have either been corrected, or

marked and allowed to pass for correction at higher levels. The correction capabilities of the keypuncher and verifier are limited to illegible and illegal characters that are observed before the document is keypunched. An example of illegal characters would be an oh (ø) which the keypunch would automatically change to a zero (0). An illegible character could be a slash (/) which looks like a one (1). In correcting these kinds of detectable errors, the keypunch and verifier rely on training and experience in the structure of the data elements. Any document that contains a blank, but requires a value would be set aside, and returned to the data checker section for action. The data checker section would then decide on the procedure to be followed. The document would be sent back to the generator or the error corrected at the checker location, according to its error priority.

The possibility of correcting detected errors after keypunch and verification requires returning the detected error record (now in machine format) to the data checker section. This is accomplished via the data output section, which has the responsibility for detecting errors introduced by the keypunch and verification process. While the output section can detect differences between the source form and the output of the keypunch, the two documents must be returned to the data checkers for the proper correction and re-entry into the keypunch procedure.

Local Computing

The local computing facility receives, as input, data in machine readable form. As such, the correction procedures are mainly computer-assisted The amount of correction possible at this location is a function of the computational power available. Other factors include the amount of reports originated and the range and depth of cross-reference and ancillary files located at the location.

Assume a full range of computational power equal to the cross-reference and ancillary files available in the

system. Then the correction procedures will be determined by the system requirements for accuracy at the management levels serviced by the computing facility.

The correction procedures at this location will, as for other locations, be either exact or approximate in nature. The exact procedures depend on cross-reference and ancillary files, which are indexed to relationships between data elements or are unique to specific data elements. The approximate procedures are based on statistical analysis of past history and on logical bounds available for selected data elements.

Exact Methods

In relying on cross-reference and ancillary files for exact correction, the detected errors are subjected to a match of known relationships. These relationships are between data elements that are correct, and those detected as in error.

The key to the correction of the detected errors is the method used in detection. For independent detection methods, the correction depends on the degree to which the erroneous data element contains the necessary relationship for correction. An example would be a structural data element code set that contains error correcting digits. The detected errors would be corrected through a computer-aided program based on the logic of the error correction coding scheme.[1]

Records that contain errors could be either corrected as they are found, or all records that contain such errors could be placed on a separate file for correction at a later time. The latter procedure does not seem reasonable since the record is needed for other detection and correction procedures involving the other data elements of the same record. Therefore, it seems reasonable to require the correction process to follow sequentially from one data element to the

---

[1] Ibid, Chapter VI, for a description of several error correcting codes.

next until all data elements of the same record that were in error are resolved.

In cases where independent detection procedures were used and the code does not contain error correcting digits, relationships will be needed for the correction process. There are several kinds of correction relationship procedures that are possible depending on the data elements involved.

Simple correction relationship procedures are where the two-data elements are in a one-to-one relationship or where one of the data elements has a very small code set relative to the other code set. The joint set relationship is composed of two data elements where one element is a modifier of the other. For example, a relationship where one code set is small compared to the other is the unit of issue of material and part number. In this case the units of issue are defined as each, set, feet, pints, quarts, square feet, etc.[1] While the part number uniquely defines the unit of issue associated with the part, the reverse is not true. But the relationship is needed, for such data are important to project the total cost of materials consumed.

Similar error correction relationships can be performed on two independent data elements, as long as there is a third code that is related to the two independent data elements. The third data element acts as a bridge between the two independent data elements. In such cases the relationship is one way -- that is, there is one data element assumed correct, or known to be correct by some procedure. The correct data element will always be the base element for correcting the other independent element, through the third element that performs the bridge between the two independent elements.

---

[1]Within the Department of Defense there are over three-million stock numbers, while the different units of issue number less than 500.

In the case of the tri-joint code set, the addition of the third data element will allow for unique correction, if such correction is possible. The third element then becomes a necessary condition for correction of two independent data elements. The third data element is not sufficient for correction, since the joint code set at the third level may not uniquely define a correct code.

An example of a tri-joint code set would be the correction of the particular manufacture of a major piece of equipment. The two independent data elements would be the plant and the manufacturer who has equipment at that plant. The data element that is related to both of these data elements is the identification code of the equipment. The equipment identification code, together with the plant, produces a plant configuration file that contains the equipment and unique manufacturer of the equipment. Such a correction could not be made between equipment and manufacturer alone since there are multiple manufacturers of the same piece of equipment. However, when the location of the equipment is specified, the uniqueness of the manufacturer is possible.

The possibilities of extending the joint code sets depend on the relationships that are available among the data elements being collected. The more relationships that can be specified, the greater the possibility of correcting the detectable errors. For example, assume a record contained n data elements, and one data element was determined correct by an independent procedure. If one other data element was related to the remaining data elements, there would be $N-2 + (N-2)(N-3)$ combinations of relationships for the data elements. This would be equivalent to $N-2$ relationships for each of the data elements. It is not feasible to perform such a large set of relationships and correction possibilities for each data element. However, having more than

one possibility does seem practical, at least from the standpoint of alternative costs and ease of performance.[1]

Approximate Methods

In using approximate methods of error correction, there are several basic schemes. These include the use of bounds, statistical averages, and statistical probabilities for determining the "best" code.

In using approximate methods, the correction procedure will generally include relationships for narrowing the possibilities available. Approximate methods must be used to select the "best" code for correcting the error. However, when correcting independent data elements, the use of bounds will guide the selection of the correction procedure.

The data element classes to which these procedures are appropriate include both the factual and judgmental. In particular, the bound techniques are employed on data such as the date, the time, and physical measurement data elements.

Within the class of procedures that relies on the concept of bounds, various decision rules can be applied. In general the amount of historical data available on the particular data element will determine the range of the decision rules.

In addition, the bounds of a particular data element are associated or related to the environment surrounding the data element. A data element may have a decision rule that is universal for all source locations. Other data elements may have decision rules that vary with the location or source of the data and a universal rule would, in many cases, not be appropriate or correct.

---

[1]There are in total, N factorial combinations, but the elimination of the two independent elements reduces the total combinations. In Chapter IV a detection procedure was developed for multiple-joint relationships where the procedure depended on adjoining pair-wise relationships. The same procedure would be used here to reduce the number of possibilities to a reasonable number.

For example, the date may have a universal decision rule for correction. However, a data element associated with a physical measurement, such as a bearing that is subject to wear, may have a different decision rule for each application. For the bearing, the decision rule would be related to past history concerning the bearing, while for the date, the decision rule would only be associated with the current date.

It is possible to increase the complexity of decision rules, but such an increase would depend on the error priority associated with the data element. The structure of the decision rules on bounds include statements concerning the possibility of transposition errors. First, the decision rule would change the characters of the erroneous code around to see if such a transfer is in the code set. If this does not provide a correct member of the code set, within the limits of the bounds, then a straight bound would be set for the data element.

As an example, consider a date entry that reverses the day and month such as fifth month, twelfth day. By knowing the current month to be December, and that there is a very small probability of a May date being processed December, it seems that reversing the day and month would correct the error.

Earlier, it was stated that the correction procedure is associated with the detection procedure. The above example shows this relationship in the following manner. Knowing the May date had a small probability of being received in December, sets a bound on the detection decision rule that tagged the May date as being in error. The bounding limit for the detection procedure entered into the decision rule for the correction. This was accomplished through the fact that the new date generated by reversing the day and month entries fell within the detection bounds. If, in reversing the two entries, the new date had fallen outside the detection bounds, the decision rule would have rejected the revised date and replaced the date with a bound date.

Another procedure for correcting numeric errors is the use of statistical methods. In particular, the mean and variance are used to compute the average or standard value of a data element. The data elements corrected by such a procedure are those numeric data elements that are in clear text. That is, the data elements are not codes themselves, but actual values associated with the object of the action. Examples include: cost of material, cost of job, price of an item, man-hours, and reliability and maintainability estimates.

As with the bound procedures, the first step in correcting the error is to look for transposition errors that would bring the recorded data element in the range of the standard or mean value. If this can be done, then the computed mean value will only be used as a method of determining the significance of the new entry, i.e., how far is the new entry from the average? If the new entry is outside the acceptable limit, then the decision rule would substitute the statistical average or standard value for the data element.

The value placed on the statistical variance of the data element can be considered a probability. The closer the recorded entry is to the average, the less chance of it being detected if it is, in fact, in error. The same is true for the revised value obtained through transposing the original entry.

The form of the probability states that the new entry has a certain chance of coming from a population where the mean is equal to the average or standard value of the data element. When the probability of such an occurrence is high, the new entry should be accepted as correct, even though there is a chance that it is in error. The method of computing how large a difference is acceptable is based on the value the decision maker is willing to accept as being in error, i.e., once in a hundred, once in a thousand, etc.

In general, the mean and variance are used in the same manner that quality control charts are used. Once a

data element entry is detected as being in error, the mean and variance provide the control limits for testing alternative values that can be obtained by permuting the recorded value. Each of the permutations has a statistical value, and those permutations that are near the mean value will be assumed more correct than those values that are further from the mean. When a data element is blank, the decision rule must rely on the average value as opposed to any permutation.

The number of permutations to be tested would not be all permutations of the number. It is generally accepted that single transpositions (pair-wise adjacent) account for a major part of the errors in transposing. This would mean a maximum of N-1 different pairs, when N is the length of the code to be permuted.

In summary, the use of statistical means and variances for error correction has two major applications. The first is setting limits on possible alternative codes that could be correct, through permutation of the recorded entry. The second is to apply the mean as the "best" correct entry when the alternatives are outside the limits of the decision rule or when the entry is blank.

The third area of correction procedures for numeric data elements, as well as an area for alpha data elements, is that of statistical probabilities. The procedure is similar to that explained under the quality control theme, but here there is no average to match against. The kinds of data elements that this procedure would correct are data elements that do not contain error correcting digits or data elements that are in clear text such as a date or price.

Statistical probability uses relationships between the different data elements as well as of exogenous data files maintained by the system. In using this procedure the first step is to eliminate as many code set elements as possible through the use of relationships. The remainder leaves a small subset of possible data element codes from which to choose the correct code. The next step in the procedure is choosing the "best" or most likely code. This selection is

based on the probabilities associated with the members of the small subset of data element codes remaining.

It must be remembered that the correct code is available in a cross-reference or similar file maintained by the system, but the relationships do not produce a unique match. That is, the relationships will narrow the search down to a select few, but the relationship by itself cannot choose from among the few alternatives remaining. In order to obtain the correct code, the decision logic attempts to discover the code the data generator meant to record. This is accomplished by analyzing the code that was recorded.

The probabilities that can be produced for such a correction procedure are of three kinds. The first is through permuting the recorded data element to obtain the different codes that could be correct. The second is through a set of probabilities obtained by analyzing the way the original element was selected during recording. A third set of probabilities can be obtained through historical data on how errors have occurred for this data element in the past.

(1) Probabilities by permutations

In developing probabilities from the permutations of the recorded data code, the density of the code set must be considered, along with the adjacent pair-wise permutations. If the set code is dense over the complete range of codes, then each position of the code may have the same number of characters. However, if the set code is not uniformly dense over all the positions of the set, then some pair-wise permutations are not valid code set entries and could be eliminated. This action reduces the number of possible permutations for estimating the correct code.

Consider a data element that is not dense; in particular, a data element that is numeric and three positions in length. The length allows for 1,000 different codes. Assume that only 200 of the codes are used, and in the following manner: the first position allows for a 0, 1, or 2 only; the second position allows all numerics except 9, as does the third position. Such a structure eliminates any

code with a 9 in any position, and all codes with an initial numeric of 3 through 9.

The reduced code set allows for a position by position check to see if the permutation would be a valid member of the code set. If a permutation was not a member of the code set, the correction decision rule would not consider the permutation. On the other hand, if the permutation was a valid member of the code set, it would be considered as a possible correction for the erroneous code.

At this point the list of alternative codes that are members of the code set are matched against the relationships that detected the data element code as being in error. If the results of the match produce a unique code, then that code is considered to be the correct code. If the results of the match produce two or more codes that would be considered correct codes, the decision rules must choose between the alternatives, on some probability scheme. If the matching produced no codes that could be considered correct, the decision rules choosesa correct code by another procedure, or assumes the code cannot be corrected.

One such method of selecting the correct code from among two or more acceptable alternatives would be to analyze the resulting acceptable permutations from the standpoint of transposition probabilities. Transposition probabilities are produced from an a priori knowledge of the distribution of characters that are generally interchanged in transcribing codes from one form to another.

(2) Probabilities obtained from the original recording list

The second set of probabilities that can be used for correcting the erroneous recorded data element code is through the techniques used to obtain the original entry. In many cases, the original code is obtained from a code book, or catalog. The organization of the book or catalog can provide information on what errors are most likely to occur through misreading of the code book. That is, recording a code directly before or after the code that was observed during the action.

Code books or catalogs will usually be in one of two sequences. One sequence is by code number which would follow numerically or alphabetically. When such a sequence is used, the correction decision rule can account for such an error and provide a technique for testing its occurrence.

The second kind of a sequence is through a functional relationship. If such a sequence is used, the decision rules become more complex. The complexity is due to the need of obtaining a list of possible correct codes during the error detection process. This is accomplished through the various relationship files used to determine the recorded code that was in error. That is, this component does not belong to this equipment, but here is a list of components that do belong to the equipment. Another is, this inventory item is not part of this order, but here are the items that are on this purchase order.

With the items belonging to the relationship, a list is generated from the catalog that contains the codes on either side of the list generated by the relationship. With this new list of all the adjacent codes, a match is attempted between the erroneous data element and the list of adjacent codes. If a unique match is obtained, the correct code is known; if a match is not obtained, the process would terminate, and the erroneous code would be marked as a detected but uncorrected error.

(3) Historical based probability distributions

The third method of obtaining probabilities for correcting the erroneous recorded data element code is through a historically based error distribution of the particular data element. Such a procedure requires that an accurate record of the kind of errors detected and corrected be maintained at the different detection and correction locations. The records would be forwarded periodically for recomputation of the error distribution. The basic characteristics of the distribution would include the particular codes that were detected as in error, and the correct code that was provided for that detected error code. If more than one correction

code was used for any detected error code, the detected error code would contain a distribution in itself.

In using the procedure, the first match would be between the new detected error and the historical error list. If a match was obtained, the distribution of corrections would determine the most likely correct code. If a match was not obtained between the new detected error and the historical error list, the correction decision rule would mark the error as uncorrected.

The three methods of obtaining corrections to erroneous data element codes through a probability distribution are not independent of other methods. Since these methods are approximate methods, they should only be used as a last resort. It is conceivable that the methods could be used together for correcting the same data element, in the sense of redundant procedures. If the three were used as redundant procedures, the probability of having the correct code if two of the procedures yielded the same code would be quite high. If two of the procedures did not yield the same code, the decision as to which code to select could be deferred to a human decision by displaying the two choices and letting an analyst decide.

In addition to the question of redundancy, the procedures could follow each other in series; that is, a successful match which yields a unique correct code was not obtained by one of the methods. The next step for the decision rule could be to try another method before assuming the error was not correctable.

The degree to which such serial correction is attempted depends on the worth of the data. As the data worth increases, more correction will be attempted before the correction process is terminated.

If the data element has very high worth, the detectable errors could be displayed for human correction, and, if possible, returned to the data generator for correction. Extreme care must be used in returning error records to the source. There is a large chance of both records

being retained in the system. Therefore, if one error is important enough to require going back to the source for correction, the inclusion of both records compounds the error problem to a higher degree than the single error.

In addition to the procedures described above for numeric codes, there is an additional statistical procedure available for alpha codes. As with several of the other procedures, the method is statistical or probabilistic and, therefore, a member of the approximate procedures. The procedure relies on the distribution of errors committed in transcribing alpha characters. It is known that not all alpha characters have the same probability of being confused with each other. For many of the alpha characters, there is little or no confusion, while for others, there are several characters that can be misinterpreted. For example, K is confused with R , X and P ; N is confused with M , and W ; and V is confused with Y and U , etc.[1]

With knowledge concerning those relationships with the highest probability of error, decision rules can be formulated which will provide alternatives to the erroneous recorded data code. The decision rules to be formulated would consider the process used in the detection of the error. Assume that the detection process detected an error at a level where the possible correct code was limited to a small subset of the data elements code set. The decision rules could be formulated in two ways.

The first way to formulate the decision rule is to consider what changes (enumerations) are required to the recorded code to match at least one of the acceptable codes of the detection subset. Associated with such changes, a conditional probability would be obtained, which would reflect the validity or reliability of making such a change. Using this change as a benchmark, other changes would be

---

[1]Owsowitz and Sweetland, op. cit., 22. In addition, see M. A. Tinker, Legibility of Print (Ames, Iowa: Iowa State University Press, 1963), for a formal treatment of the complete subject.

made and the conditional probabilities obtained. The result is a list of acceptable codes, with each code an associated probability that reflects the chance of such a code being derived from the original code. The decision rule would then pick the code with the highest probability as being the correct code. The decision rule would also contain a threshold value. If none of the probabilities was above the threshold, the error code would not be corrected, but marked as detected and uncorrected and possibly displayed for human correction.

The second way to formulate the decision rule would be to consider only the conditional probabilities associated with the changes. That is, the decision rule would first change the recorded data code in a step-wise fashion, checking at each step to see if the new code matches an acceptable member of the subset.

The procedure for the step-wise changes would select as the first data character to change the highest probability contained in a "confusion index." That is, the "confusion index" has ordered, by probability, the characters that have the highest chance of being in error. If that change does not match an acceptable member of the subset, the next character with the highest probability of being in error would be selected. Each time a new character is selected, the original character is returned so that the changes are made one at a time until all characters of the code have been changed and replaced if a match did not occur.

When the first match is obtained, the procedure evaluates the probability of such a code being obtained from the recorded code. The decision rule accepts or rejects the new code on the basis of the probability and assigned threshold. If no match is obtained, the procedure terminates and the data element code is marked as an uncorrectable detected error.

The main difference between the step-wise decision rule and the enumeration decision rule is the amount of

combinations that needs to be checked. Also the enumeration decision rule will obtain a correct code although it may be below the threshold, while the step-wise may not provide a correct code.

This difference is seen by the fact that the enumeration decision rule takes the erroneous data code and transforms it into the acceptable codes of the subset. Each transformation has an associated probability, and the new number has a probability. Since there are no limits to the number of individual transfers a code can go through, all the characters of the code can be transformed. However, the probability of some of these changes is very small, and if the changes cause the probability to fall below the threshold, none will be accepted as the correct code. The chances are better that a code will be selected by this process than by the step-wise process, since only one character can be changed.

To summarize, the correction methods available at the local computing location are mainly computer-aided programs. The degree to which the procedures are used, as well as the complexity of the procedures are primarily a function of the computing capability available. In addition, the master and ancillary files available, and the data worth associated with the different data elements will influence the degree of correction performed.

There are exact and approximate correction procedures. The former are based on joint code set relationships which combine data elements collected in the system as well as data available from master and ancillary files. The latter depend on bounds, statistical inference, and probability estimates. When a choice between exact and approximate methods is available, the former should be considered first, because of the uncertainty associated with the approximate. However, two factors should be considered: first is the cost of accomplishing the exact correction as opposed to the approximate procedure, and second the level of accuracy required of the data element.

The Central Computing Location

The correction procedures available at the central computing location are similar to those available at the local computing centers. Differences between the computer-aided programs are due to the computer capability available at the two locations.

In addition, the local computing facility may not have the necessary variety of input data to perform all of the correction functions. That is, the data may be so structured that the data elements required for certain relationships in the correction procedures are not available at the local level. Only through a file maintenance program generated by the central computing facility can all the relations be defined. If such a file maintenance program is available, it suggests that a degree of redundant error detection and correction is taking place. In this case, the redundancy should be controlled, or at least its cost known and monitored for possible reevaluation of responsibilities.

A second advantage that a central computing facility may have over the local computing facilities is the use of exogenous data. The basic fact that there is a central computing facility suggests that some data are transferred to the central computing facility without first going through the local computing facilities. In particular, this would include data submitted from organizational activities above the local computing as well as those above or on the same management level as the central computing facility.

In an on-line real-time information system, there may not be any local computing facilities. All such functions then would be provided by the central computing facility. One additional correction procedure available under an on-line real-time system involves the use of the data generator, i.e., sending the record back to the generator for correction. Such a procedure is much simpler and "cleaner" for this system than for the general batch processing system. The procedure requires a display device at the data generator's

location. Through a set of standard questions the data generator is asked to reestablish the necessary relationships required to verify that the data submitted are correct. These questions are asked even though the detection procedures have suggested that some of the data are in error.

If the data generator cannot establish the necessary relationships to show the data to be correct, computer decision rules will require the data generator to change those data still considered in error. The requirement that the data generator establish the relationships stems from the kind of data elements being corrected. If the elements have correction procedures that are exact, the relationships are automatic. If the correction procedures are approximate, the decision rules governing the error are statistical. Therefore, the procedure has some probability that a correct answer could be outside the statistical limits placed on the data element.

The possibility that a correct answer could be outside the established limits becomes evident through the questioning of the data generator. Such a procedure has two possible benefits: the first is continuous training and a better system understanding by the data generator. The second is the ability to verify and update the decision rules associated with the data element.

The data generator benefits through association with the questions. The degree of complexity of the questions, and their logic, will require the data generator to have a more than general knowledge of the subject matter. The logic, however, would not be so complex that if the reported action was actually accomplished according to system procedures, the data generator would not be able to answer the questions.

This, in itself, results in a secondary system benefit. The data generator would be more careful in the original reporting, and he would not try to "beat the system." Especially if he knew that all of his answers were being checked through such a system of error detection and correction.

The second benefit is to the system managers trying to improve the efficiency and productivity of the system. In providing the original decision rules associated with the different detection and correction procedures, there is the possibility of some obscure relationships being omitted. The reestablishment of the basic relationships as well as others that might have been omitted will be helpful to the improvements of the system's error detection and correction procedures.

Data Analysts Locations

In error correction, the data analysts can perform two kinds of functions: the first is the correction of products associated with the system, and the second is correction of the data elements which the computer-aided decision rules were not able to correct.

Corrections Associated with the Products

The data analysts performing this function could be better classified as information analysts since the procedures deal with the correctness of the information products. The products must be correct in that the data elements used to produce the reports have meaning in an operational or management sense. To be more specific, there must be a logical relationship between the data and the intended use of the report. The logic should be well established in a known management requirement.

The analyst's job requires knowledge of the management world to which the information is being applied, as well as detailed knowledge of the workings of the information system. The analyst in performing this duty, acts as the bridge between the information system and the output products for management use.

The range of correction procedures that the analyst is able to perform depends on his management knowledge and the area in the organization where the particular functions are performed. In general, the data analyst should be able

to correct all output reports. This includes corrections associated with programming errors or data errors that are obvious to human observation as well as more specific mathematical errors, magnitude errors, and unit or scale error.

A second kind of correction involving the products would be errors in the use of data elements. Here the procedures require relationships between the intent of the product and the proper data to use in preparing the product. In this area the data analyst works closely with the information user who may have only a superficial knowledge of the data available within the system.

The correction procedures come about when the analysts observe that there is a more efficient way of either displaying or obtaining the information required by a particular user. In bringing this fact to the attention of the user, the user and analyst together can better meet the actual information requirements of the user. Such an interchange between the user and analyst will provide two benefits: (1) an improvement to the user in his decision making responsibilities, and (2) an improvement in the system responsiveness to user requirements.

Correction of Data Elements

In some instances the rules required to correct the detectable error are not logically possible. While some errors could be machine corrected, an excessive amount of computer time would be required to test all of the relationships. In many of these cases, a scan by a knowledgeable analyst of the detected errors organized by some basic relationship will give the data analyst the necessary information to correct a significant portion of the errors. Examples include such errors as alpha for numerics, data record field shifts of clear alpha text, and left and right justification of data fields. While such errors may be correctable by machine through elaborate decision rules, it may be faster and simpler to use human correction.

The basic correction procedure available to the data analyst in performing this scanning technique is human logic as opposed to machine logic. By scanning an array of data, one can observe relationships that the computer would miss by performing a serial match of the error data. Examples of such array scanning would include detectable errors in such elements as part numbers which can include dashes, slashes, and alpha prefixes and postscripts.

When these special characters are missing from a part number, the computer would identify each change as a separate part. A printed array of such numbers could be easily corrected by the analyst since he could scan both vertical and horizontally in looking for relationships.

In general, there are other such detectable errors that are more easily corrected by the analysts than the computer. Another whole class of errors are those of written text or vocabulary. In many systems clear text is used to describe selected data elements. When clear text is used, the data generators may be given freedom as to how it should be prepared for the system. If free text is used, abbreviations are not written in the same manner and not all writers spell correctly. To include within the correction programs a large thesaurus to account for all the combinations of abbreviations and misspellings would require a very large and complex program. For such corrections it is much simpler to include a small thesaurus of the most common abbreviations and misspellings. The small thesaurus would correct a major portion of the detected errors, and allow the analysts to correct the remaining errors.

The use of the analysts for correcting errors should be considered as an extension of the computer-aided programs. Some advantages are unique to the computer and some are unique to the analysts. Each should be used to its best advantage. The computer has the advantage of processing a large amount of data very quickly given that the data are homogeneous and the same general decision rules can be used. On the other hand, the analysts can interact with exceptions

and variations to the data that would be both hard to program as well as difficult to predict. The man-machine interaction would have the computer correcting the errors through the common, easily programmed correction decision rules. Man would correct a much smaller subset of errors, which are not predictable and most difficult to program.

Information Users

Information users, like data analysts, perform error correction differently than the other correction locations. The main objective of the information users is to interact with the basic data through the information. The user then integrates the information provided by the system with other external information that will affect the current operation or future plans of the organization.

As the information user performs his basic objective of interacting with the information system, the user can provide an error correction function. The error correction capability is available through the information displays used by the information user. The information received by the user is either aggregations of the basic data or mathematical algorithms using the basic data. While the information user does not observe the actual data collected, only the information derived from the data, the user is nevertheless acquainted with the data elements used to produce the required information.

The errors available for correction by the information user are two kinds. The first type of errors are associated with information products of the system The second kind are detectable errors not corrected, but associated with the products of the user. In both cases the user acts as an external audit of the information system.

In correcting errors associated with information products, the user acts in the same manner as the data analyst. However, the user has more knowledge of the information needs associated with his responsibility. With this additional knowledge, the user's role in error correction is

that of information correction. That is, he is responsible for obtaining the correct data elements and correct methodology for transforming the data into usable information.

The second role of the information user is to correct detectable errors presented in his output. Since the user does not generally see the actual data, the error correction is via the information presented. This means the correction is more general than that of other correction locations.

In using the information, the user has some expectations about what the analysis of the information will show. If deviations from this expectation begin to show up, the user will question the data behind the information. In this manner, the data analysts are alerted to the problem. One solution to the problem may be erroneous data. While erroneous data may be one solution or cause of the variation in output, the information users have the ability to search out all the causes of such variation to improve the accuracy and usefulness of the system.

The level of the information user plays an important role in his ability to correct data through the information products. In general, the lower a user is in the organization, the closer he is to the actual data. That is, the less aggregate the information product is for his management responsibilities, the closer the user comes to performing the same role as the data analysts in the error correction process. The opposite is true when the user is high in the organizational structure; the less direct correction the user is able to perform.

Irrespective of the user's location, an informal role can and should be performed by the user in the error correction process. Whether the procedures are direct correction as in the case of the lower organization users, or indirect and by inference as with the higher organizational users, the user participation is required to complete the error detection and correction process.

Summary

Error correction procedures are available at all locations within the information system. The procedures to be used by each location can be separated into four general classes.

The first class of procedures is that which has direct association with the source form and the data generator, using the knowledge of the data generator and system procedures. The second class involves the source forms and their interface with the system hardware, beginning with the transfer of the data from source documents into machine readable form. The procedures available for this class go beyond those available at the first class by adding some computer-aid-programs, cross-reference, and relationship files to the procedures.

The third class goes beyond the second by adding more computer-aided-programs and the complete range of cross-reference files for use in "exact" correction. In addition, the third class includes the methodology for performing approximate corrections based on statistical inference and decision rules concerning error priority and data worth.

The fourth class of correction procedures is procedures associated with the system products. While some of the procedures are associated with the data directly, several are associated with the information content. The latter provide correction to the information products as opposed to the basic data elements collected. The correction of the information products by the users provides feedback to the other data correction locations. This enables the system to improve the correction procedures to the basic data as well as providing improvements in system performance to the users.

The error correction procedures discussed in this chapter provide a framework into which detailed detection and correction procedures can be established. In addition, the concept of error priority was described as a necessary ingredient in determining how much correction should be accomplished.

The one remaining ingredient, cost, is the subject of the next chapter. The relationship between cost, priority, data worth, and error detection and correction procedures should form a set of principles. These principles will enable the system manager or system designer to provide the system users with information that is more accurate and more usable than ever before.

CHAPTER VI

THE ECONOMICS OF ERROR DETECTION AND CORRECTION

Introduction

Cost is a major concern in performing any function of an organization. The real costs are both direct and implied. In many cases the costs of participating in a venture does not follow what would be considered good business practices. These costs are charged to something referred to as the "cost of doing business." Such a term is equated to an utility which states that the lack of such participation may be more costly to the organization than the cost of participating. The cost is not from the "good" that will be derived from the venture, but from the "bad" that may result if one does not participate.

In many cases computer based information systems fall into the same category. That is, beginning a management information system which is computer based and will require a total systems approach to the organization.[1] The major cost of such a system falls under the account of the "cost of doing business."

A part of the cost in developing information systems is the cost of detecting and correcting errors in the information system. There are two major costs of errors: the first is the actual cost of performing the error detection and correction. The second cost, which can be far greater than the first, is the cost associated with not doing the necessary detection and correction. If emphasis is placed

---

[1]John Dearden, "How to Organize Information Systems," Harvard Business Review, (March/April 1965), p. 65.

on having the proper amount of detection and correction, then the cost associated with not doing the necessary detection and correction can be eliminated or greatly reduced.

There are many examples of elaborate information systems which collect a great deal of data. Yet the information derived from the data precludes the use of the information in the decision process. This is mainly because the system as originally designed provided one level of management with information designed for use by another level.[1] In other cases, the systems are allowed to wander through the organization with little top management direction.[2]

Regardless of whether an information system is operating at the right level with the right direction, as the error content of the information becomes unacceptable, the system ceases to function. Very seldom does the system cease to exist. When the system is not functioning properly, it becomes a burden to management and an added source of cost to the organization.

To reduce the chance of an information system becoming obsolete via the data accuracy problem, this chapter will outline some of the variables associated with the process of error detection and correction. In particular, the chapter will describe classes of error detection and correction procedures and the variables that account for the cost in implementing the procedures.

Information systems can take many forms as to hardware and complexity of operation. Therefore, no attempt will be made to associate numbers to the cost of the variables -- only the relationships will be described. Each designer/manager will be required to determine the actual costs as they apply to their particular system.

---

[1]Dearden, op. cit., pp. 133-134.

[2]John T. Garrity, "Top Management and Computer Profits," Harvard Business Review, (July-August 1963).

The Elements of Costs

The previous section suggested two major kinds of costs associated with the error detection and correction procedures. The first is the cost incurred by the system itself. These are the fixed and variable costs of having an information system. The second are those costs than can be incurred by the organization through erroneous and improper use of the information in the various decision making processes.

Fixed Costs

The fixed costs associated with error detection and correction are defined as those costs incurred in maintaining total system integrity, irrespective of the amount of formal control over data accuracy. Included in these costs are those one time expenditures (initial investment costs) required to obtain a given error detection and correction capability. Examples of such costs include internal audit procedures, system analysis of the information requirements of the users, and the cost of computer hardware allocated to the correction and detection function. Examples of one time costs would include initial computer programming development, file conversion (if solely for error detection and correction), and the cost of developing reference manuals and detection devices.

Variable Costs

Variable costs are those that are directly associated with the continuous development and operation of the error detection and correction function. Examples include increased personnel costs, computer programming changes, file maintenance, continuous training and manual updating and distribution costs. Increased computer operating costs and the purchase of additional computing capacity that varies with the degree of detection and correction are further examples of variable cost. However, if all of the increased capacity will be utilized solely by the detection and correction function, then its cost becomes a fixed cost.

On the other hand, if existing capacity is available the cost would be a variable cost computed as a machine hour cost against the detection and correction procedure. In either case, there is a computer cost to this procedure, and it should be borne by the system. The same is true for reference manuals that are placed outside the technical areas where they were designed to operate. The extra cost of distribution and printing should be considered part of the cost of error detection and correction.

The Components of Cost

In the aggregate, the costs can be distributed to: (1) personnel, (2) equipment and communications, and (3) data requirements of the procedures. Requirements for personnel and equipment are easily understood. Communication requirements include all of the reference material as well as the manuals and methods of communicating either the information needed for the procedures or instructions concerning the procedures. These are in addition to the usual meaning of communication as a means of transferring data from one location to another location. The data cost includes all the file conversions, data file generations, file maintenance and data collection needed for the various procedures.

In each error detection and correction procedure the three costs mentioned previously are present. The elements that comprise the three costs will vary with the procedures from the simple to the complex. In some procedures the cost will be depicted by data checkers, desks and working space and catalogs, while for other procedures the cost elements will include programmers, computers, tape files, and communication lines.

The Classes of Error Detection and Correction Procedures

In order to determine the cost of the error detection and correction procedures, the necessary resource elements

used in the procedures need to be described. While it is not possible to describe the actual cost associated with the procedures, it is possible to delineate the resource elements that comprise the procedures. These resource elements are then the candidates of additional costs to the organization.

Rather than outlining the elements for each error detection and correction procedure, it seems reasonable to discuss the procedures through a classification scheme. The classification scheme that seems most appropriate is one that separates the procedures by the major kinds of resource elements used in the procedures.

The classes that will be used are:

1. System-manual Procedure Class

   These are procedures primarily instituted through system manuals. This class is concerned with the development and up-grading of more precise methods and procedural techniques at all levels of the operating system. In general, the greatest effort is placed at the lowest level of the system where the data are generated.

2. Manual-visual Procedure Class

   This class is composed of data checkers and similar personnel for manual-sight verification of the data relying on system knowledge. In addition, the use of catalogs and reference material would be available for detecting and correcting errors through manual techniques.

3. Manual-EAM Procedure Class

   Included in these procedures are internal audit procedures at different locations within the systems such as at the keypunch location. In general, these procedures involve the use of the source form and machine readable data as a check on the accuracy of the data being forwarded. The results of the audit procedures are used to develop new or re-emphasize current error detection and correction procedures. In addition, there are procedures that involve the capability of the EAM equipment in data admissibility and validation checking.

4. Computer-aided Validation/Admissibility Procedures

This class of procedures uses a computer program, internal decision rules and the machine readable records. That is, relationships between data elements of the same record, and simple independent checks about the data elements such as structure and composition are of prime importance.

5. Computer-aided Statistical Procedures

This class of procedures uses either statistical inference or probability techniques for estimating the presence of errors. Correction is also accomplished by statistical procedures where the errors were detected by these or other procedures.

6. Computer-aided Table Look-up Procedures

These procedures require table look-up files for use in making comparisons. Here the files are maintained within the program and updated through changes in the program and its internal decision rule logic.

7. Computer-aided Master File and Cross-Reference Table Procedures

These procedures use master files and cross-reference files maintained by the information system in the error detection and correction function.

a. Single relationship master file look-up procedures --

These are procedures that require the use of master and cross-reference files. However, the depth of detection and correction is limited to independent and simple joint code set relationships.

b. Multiple relationship master file procedures --

These are procedures that require the use of master and ancillary files to detect or correct complex multiple data element relationships.

Resource Elements

Not all elements associated with the error detection and correction procedures are easy to quantify. In many cases the elements include such resources as training,

adaptability and desire on the part of the system personnel. For these cases, the explicit identification of the elements will be useful to the system designers and managers. Where the elements can be quantified, the system designers and managers will be able to associate the cost of that resource element to the system environment in which he is working.

Elements of System-Manual Procedures

The resource elements of this class of procedures are mainly one-time costs associated with implementation, or for on-going systems a revision effort to the system manuals. While the procedures are simple, they do require formal documentation and testing before implementation. The resource elements associated with this class of procedures include:

1. Training -- This concerns the training of data generators and supervisors in the formal procedures that are to be implemented. While any information system will require training, the introduction of formal error detection and correction procedures will increase the initial training requirements as well as periodic refresher and modification training.

2. Procedural integration -- Integration at the data generation locations includes the actual procedure writing, examples or actual forms or displays for detecting the errors, the procedures for correcting the detectable errors, and manual preparation. The one time cost of providing explicit, yet simple procedures, will provide the basic building block for the formal error detection and correction procedures throughout the system.

3. Error detection and correction aids -- Of prime importance in providing adequate training and indoctrination is the development of aids. While the aids could be discussed under training, they are more in the nature of independent aids that each location should develop for improving their detection and correction capability. Since the aids are tailored to specific locations, basic guidelines should

be established by the system designer or system manager to insure consistency and coverage.

4. Communication resources -- The basic requirements for communications are within a data generation location. The more compact each location is, the less resources required to implement and perform the error detection and correction function. When the system procedures involve source data recording on data collection forms, the communications requirements will include the necessary devices to collect the data within a data generation location.

In system applications where automatic source data collection is used, there is an additional requirement for communication back to the data generators. This feedback from the supervisor will provide the means of informing the data generator of errors detected and corrected by the supervisor.

5. Equipment resources -- Equipment resources, like communication resources, depend on system complexity. For data collected through source data forms, the equipment requirements are minimal. The resources that are available for system operation will, in general, be sufficient to provide the additional capability need for the detection and correction procedures. The use of automatic source data collection procedures will, however, increase the equipment requirements. The increase will be through devices that will enable the recorder as well as the supervisor to enact the required procedures.

Manual-Visual Procedures

Manual-visual procedures overlap those of the system-manual procedures through the role of the supervisor and data checker. The resource elements associated with these procedures would include:

1. Training -- Providing the necessary training program for the data checkers. In general, this could be approached as a two-phase training plan, conducted at the data checker work area. The first phase would include system training as to the objectives and importance of the

system and the specific job of error detection and correction. The second phase of on-the-job training would continue until a required level of competence was reached. This resource element would be a continuous cost to the system. The cost would vary in relation to the data checker turnover rate and increased system workload.

2. Procedural integration -- The cost of procedural integration for this class requires the preparation of detailed instructions to the data checkers as to the proper use of each procedure. Such instructions would include the detection and correction methods to be performed within the data checker location. Similarly, detailed instructions on the interface with the data checkers in returning detected errors for correction to the data generators are necessary. Since the system is dynamic, the procedures as well as the instructions concerning the procedures will be changing. This will cause a continuous cost in maintaining and updating the procedures at the data checker location.

3. Detection and correction aids -- Resource requirements for aids to be used with this class of procedures involve such visual aids as check lists, table look-ups, tailored tables, reference documents and catalog files. In addition, the use of hardware aids such as templates, desk adding machines, and calculators would be needed. The initial distribution of aids would be part of the fixed costs; catalog material would be a continuous cost to system operation.

4. Communication resources -- The communication resources are related to the geographic location of the data checkers to the data generators since the greatest interaction is between these two locations. The number of documents returned to the data generators and the degree of corrections provided by the data checkers will affect the size of the communication network. Response time will be affected by the distance between the two locations and the specific communication equipment used.

5. Equipment resources -- The basic equipment requirements are those of office and administrative space necessary for the function. The amount of such equipment will vary with the volume of input, the number of data checker stations at each location and the number of locations.

6. Personnel -- The need for data checkers is solely a requirement of the error detection and correction procedures and a variable cost to the system. The required number of checkers at each location depends on the depth of detection and correction required as well as the volume of data received. The number of documents that can be processed per day, divided into total expected documents to be received per day, plus allowances for administrative functions will provide an estimate of the daily number of data checkers.

Manual-EAM Procedures

Resources for the procedures of this class are generated through two basic requirements. The first requirement is the resources needed to perform part of the error detection and correction audit function. The second are resources needed to perform error detection and correction at the interface of the data collection form and machine readable processing equipment.

1. Personnel requirements -- The personnel needed for the procedures of this class are of two major kinds: those technically skilled in data analysis techniques to perform the audit function, and the personnel capable of operating and implementing the procedures requiring the use of EAM equipment. In the latter case, most of the personnel may be available and all that is required is training and a small augmentation of that force. The same availability may not be true for the data analysts. The analysts' cost would be charged to the detection and correction function.

2. Equipment requirements -- System equipment requirements are determined by the volume of data records that are processed by these techniques. In general, the use of EAM equipment for such detection and correction is faster

than manual procedures, but many times slower than computer techniques that would perform the same function. For this reason alone it would seem profitable to use the excess capacity of the system equipment already on hand, and limit the depth of error detection and correction to that capacity. However, if current EAM capacity is not adequate, the additional cost is charged to the detection and correction function.

3. Procedural development -- Resources will be required for the initial development and implementation of the procedures. A continuing cost will be associated with maintaining and updating the various procedures. In connection with the audit procedures, additional resources will be required to develop, test, evaluate, and implement new methodology and audit procedures into the system.

Computer-Aided Validation/Admissibility Procedures

This class of procedures provides the initial computer-aided programs to the detection and correction process. The resources required for this class of procedures include:

1. Personnel -- The personnel resources include programmers, data analysts, system analysts and operating personnel. The initial cost of programming personnel will be in writing, testing and debugging of the basic admissibility relationships. Continuous costs will include the computer operating personnel to run the programs and programmers to maintain and update the program in accordance with system changes. Additional continuous costs would include the data and system analysts' functions.

2. Equipment resources -- The major cost associated with equipment resources required by these procedures is the computer and its components. The computer hours required to perform the basic admissibility checks of the data depend on the detection and correction that have preceeded this class as well as on the volume of data that is being processed.

3. Communication resources -- There are two major kinds of communication resources expended in this class of procedures. The first is communication in the form of feedback to the data generators concerning the kinds of error they are committing. This will be a continuous cost and vary with the detail and amount of the error feedback reports. The cost of preparing these reports would include all of the functions concerned with the printing and distribution of the reports.

The second major cost is associated with the actual transfer of the data to the computer. If remote terminals transfer data to a central computer, there will be an increased capacity requirement for the feedback of message traffic between the computer and the terminal locations. The feedback traffic will be determined by the depth of the validation and admissibility checks, and the decision logic concerning the kinds of errors returned to the terminal for correction by the data generator.

Computer-Aided Statistical Procedures

This class of procedures involves the use of statistical techniques for detecting errors as well as the use of statistical techniques for correcting detected errors. The resource elements of this class of procedures include:

1. Personnel resources -- The personnel resources include such skills as programmers, system analysts, data analysts, analytical statisticians, computer operating personnel and statistical clerks. The programmers required for these procedures should be knowledgeable in mathematics to the extent that they are able to communicate with the statisticians who are preparing the specific procedures. To some extent, the same is true for the system and data analysts. That is, they should understand the meaning of the results, and be able to implement changes to the programs when the results indicate such a need.

2. Equipment requirements -- The basic requirement is for the necessary computer capacity and configuration to

provide the degree of logic and computational power to carry out the necessary procedures.

3. Communications requirements -- This requirement is similar to the requirements and conditions described for the validation/admissibility procedures, with the added requirement of a possible further distribution of feedback reports. The additional requirement of the feedback reports would provide to lower level error detection and correction locations, the new values of the variables to use in their detection and correction procedures.

4. Procedural development and integration -- The successful execution of these procedures requires test and evaluation before implementation. While the resources required to perform these functions are covered by the personnel and equipment costs, there is a need to delineate the importance of the integration of these procedures.

The importance of the statistical correction procedure, as a tool for use when detection was provided by a more exact method, should not be overlooked. It is very possible that certain kinds of errors would be detected at a lower level in the system, which could not make the correction. In such cases the statistical correction procedures may be the only way of obtaining an acceptable value for the erroneous data element. It is for this reason that in-depth analysis of the various procedures should be undertaken to insure the best possible integration of the procedures to be used by the system.[1]

5. Data requirement -- Data requirements include both the data needed by the procedures as well as the data generated by the procedures. The use of many statistical as well as probabilistic models requireshistorical data as a basis from which to generate future values or average cumulative current values. To implement the procedures such data

---

[1]This is not only true for these procedures, but for procedures of all classes. The statement appears here since this is the first complex class of procedures that can provide corrections to elements that were detected earlier by other means.

constitute an absolute requirement for the system. The amount of data required depends on the past records as well as the depth and range of use of these procedures. Once the procedures are implemented, the data requirements are self-generated as output of the procedures. The resources required for the output involve a data handling system for the storage and retrieval functions. Such a data handling system should be part of the overall data management structure of the information system.

Computer-Aided Table Look-Up Procedures

These procedures are used to provide depth to the error detection and correction function. The depth is provided by tables contained in the computer program that compare new data entering the system to known correct data contained in the tables. The resource elements for this class of procedures are similar to those of the other computer-aided procedures, but vary in the skills of the personnel and data requirements.

1. Personnel requirements -- The skills required for some of the programmers, data analysts and system analysts must reflect a knowledge of program list techniques and table sequencing.

2. Data requirements -- The data requirements for these procedures are of two kinds. The first is the data associated with the data element code sets that are included in the tables. The second are the data generated by the system and data analysts to provide joint relations between the data elements. In the former case, the data are available and need only be sequenced according to the program requirements. However, in the latter case, the data will need to be generated and tested for consistency before it is incorporated into the procedures.

Computer-Aided Master File and Cross-Reference Procedures

This class of procedures is by far the most complex and will require the greatest utilization of resources. As with many of the procedures, this class can be implemented in phases. The planning for these phases should be well developed so to minimize the possibility of inefficiencies through programming, data handling and computer operation. The procedures will require similar resources to those of the other computer-aided methods previously described. In addition, these procedures will require:

1. Personnel requirements -- Additional technical personnel familiar with the structure, availability, location and content of data files obtained from outside the information system.

2. Computer requirements -- The increased computer time required to perform these procedures depends on the data volume and the degree to which such procedures are developed. In very large information systems, one pass of a master file may account for several hours of computer time.[1] Attempts should be made to be as accurate as possible on the size, use and organization of such files to insure optimal use of available computer resources.

3. Communication requirements -- Since the procedures of this class are concerned with computer-aided methods, the amount of communication required is related to the degree of centralization of the system. The more centralized the system, the greater will be the requirement for communications between all of the lower activities and the central computing facility. This is especially true in a real-time on-line system where each data generator may be connected to the central computer.

In the conventional batch processing information system, the communication requirements would be less for the

---

[1]The Social Security computer facility has one master file that is 2,400 tape reels. This is passed once a month.

batch processing than for real-time on-line. This is true as long as all of the master files were located at the central facility. However, if the master files were decentralized, with updating responsibility still located at the central facility, the communications requirement would increase. The increase would be in proportion to: (1) the number of locations where the files are located; (2) the frequency of updating the files; and (3) the volume of data transferred between these locations and the central facility.

4. Data requirements -- The use of master files places a large requirement on obtaining and maintaining the data base. In addition to the requirements of organizing the data in the master files, there can be large resource requirements for file conversion and file maintenance. Experience has shown that conversion and maintenance of master files can be extremely costly.

Many of the files needed for the detection and correction function are dynamic. In dynamic files there is a constant problem of updating, historical retention and audit. That is, knowing what file changes were made and when the changes occurred. The need for good data base management cannot be over-stressed. For this class of procedures, the two largest costs are contained in the files and the computer, both of which are continuous costs.

There is a large variation in the degree to which this class of procedures can be used. In the case of a general batch processing system which receives the data by way of source forms, other procedures should be attempted before a large commitment is made in these procedures. However, in an on-line real-time system with remote terminal data input, there is little choice but to allow the computer to perform all the necessary error detection and correction procedures. There is one advantage to the on-line real-time system which helps to minimize the cost of detection and correction. Most of the detection is provided by the computer, but the correction is performed by the generator at the terminal. In this manner, the computer-aided programs are mainly

interested in providing or maintaining the detection process which is somewhat easier to perform.

The Economics of Detection and Correction

It has been shown that error detection and correction procedures can range from the very simple to the most complex. The degree to which detection and correction procedures are employed is determined by the system's requirement for accuracy and the resources available to develop and implement the required procedures.

There are various trades-off between the requirements for accuracy and the costs of providing the accuracy. It seems reasonable to assume that these relationships would be similar to that displayed by Figure V.

Figure V has two unique features. The first is that the left tail cuts the abscissa before it reaches zero. This means that not all data collected will be in error, and that without spending any resources for detection and correction, some part of the data will be accurate.

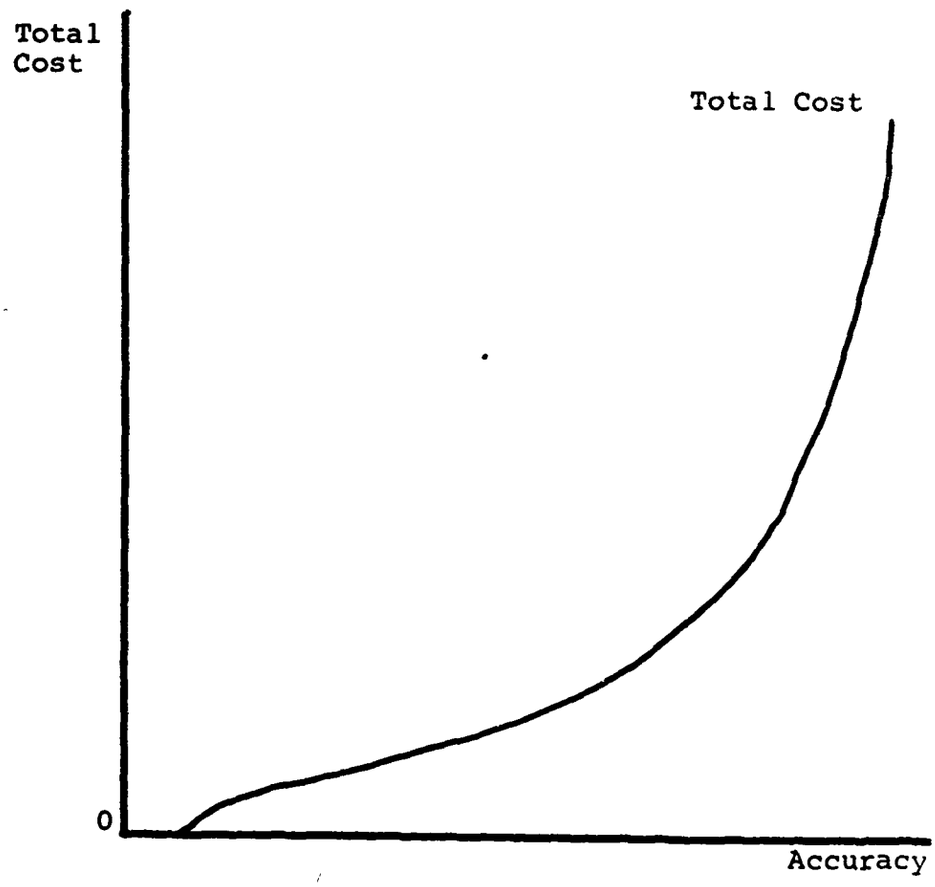
It is not possible to determine the value where the cut actually occurs, but one would doubt that the accuracy of the data would approach the system requirements. In addition, it should be noted that good system design would include error prevention procedures which could move the intercept further to the right.

The second point is that, as the system attempts to reach 100 percent accurate data, the cost of such accuracy rises very quickly. That is, one additional unit of accuracy costs more as the system approaches 100 percent accurate data than the one additional unit of accuracy costs at the lower end of the curve. Such a curve suggests that system data accuracy should be set somewhere between these two extremes.

In addition, the shape of the total cost curve of Figure V suggests that there are increasing as well as decreasing returns to scale. Again, at the lower end of the curve, each additional unit of increased accuracy adds less

Figure V

RELATIONSHIP BETWEEN ACCURACY AND COST[1]



[1]This curve as well as the following curve are conceptual in nature, but seem to reflect current literature concerning the economics of information, i.e., a polynomial as opposed to a straight linear relationship.

to total cost than the previous unit. This is due to the fact that each unit of the resource adds more to the accuracy than does the previous unit of resource.

In the upper end of the curve the opposite is true; here each additional unit of resource adds less to total accuracy than the previous unit. One reason for this is that an input resource may be fixed, limited, and indivisible.

The middle section of the cost curve suggests that there may be some range where there are constant returns. That is, an additional unit of the resource adds the same amount to total cost as the previous unit of resource over some range of accuracy.

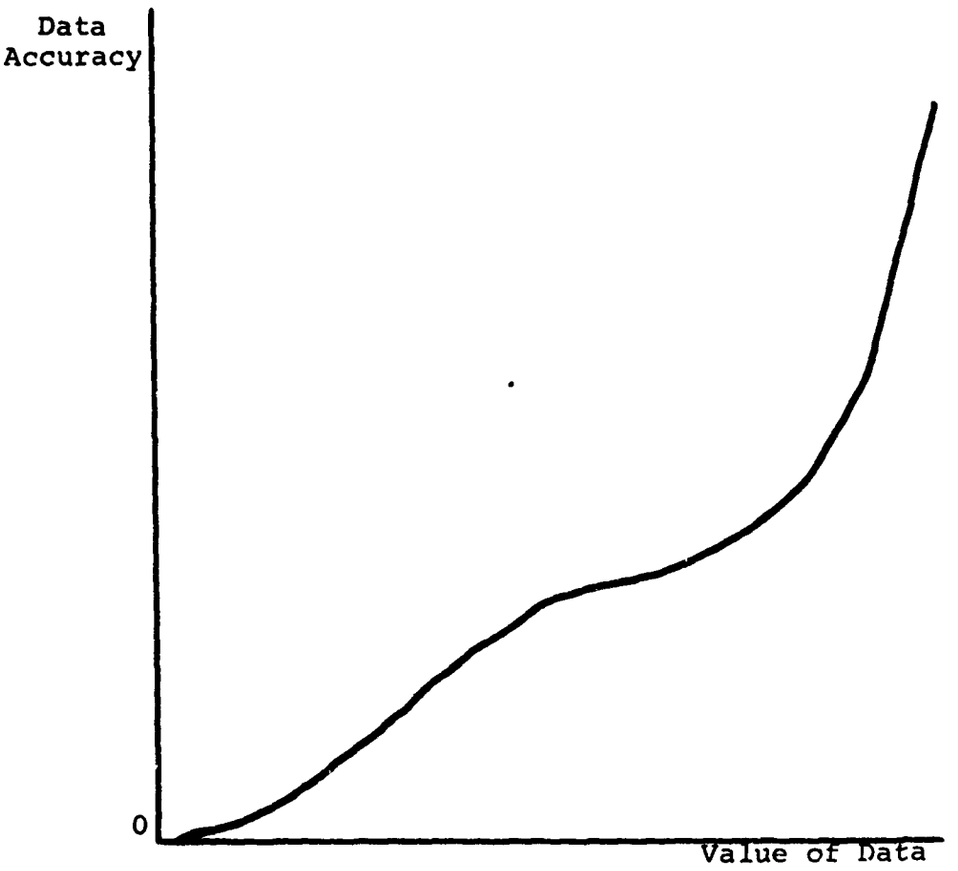
The decision as to the amount of data accuracy to provide should be based on the economic pricing of the factors required for obtaining that accuracy. A major factor is the demand for such accuracy by the system users reflected through the worth of the data. That is, what price per unit of accuracy are the users willing to pay for accurate data? Such a costing procedure suggests that the system accuracy requirements should be set at a level equal in price to the value the users place on the accuracy of the data in their decision process.

Figure VI displays a curve that may reflect the relationship between the value and accuracy of data. In Figure VI the value of data to the user and the accuracy of data are positively related. This means that the higher the value of the data, the more accurate the decision maker wants the data for use in his decision process. The shape of the curve in Figure VI tends to become asymptotic at a high value. This suggests that as the data approaches a maximum value, data accuracy increases rapidly for a small change in data value.

In addition, the curve may not be continuous as depicted by Figure VI. The curve may be similar to a step function, although still monotonically increasing, with a value range for each level of data accuracy. If the latter

Figure VI

RELATIONSHIP BETWEEN VALUE
AND ACCURACY OF DATA[1]



[1]See footnote for Figure V.

is the case, the users may specify their value requirements through such generic terms as bad, fair, good and excellent. In either case, the relationship between the value of the data to the users and data accuracy needs to be established.

As an example, consider the situation where inventories are stocked according to a technical estimate as opposed to actual demand requirements of a reporting system. For one system the difference between the technician's estimates and the requirements derived from the demand data reporting system differed in favor of the demand data system by a factor of two.

The value of accurate data was determined by the difference between the amount stocked by the two methods to meet the system's inventory demands. By using the demand data system as opposed to the technical estimates, the dollar amount approximated three-million dollars per inventory location. In this specific case, there were forty-one inventory locations.

Assuming that a relationship between value and accuracy of data can be established with the information users, the relationship between accuracy and cost could be developed by the system designers. With these two relationships, the link between what is needed by the users and the costs of adequately providing that accuracy can be obtained. The relationships will show the cost of providing the desired degree of accuracy required by the users, as well as the complete range of the costs associated with all levels of accuracy. Given the complete range of costs, the system designer can provide to the users the cost of any accuracy level. In fact, when constrained by resources, the designer can provide to the users the resulting accuracy levels obtainable from the available resources.

There are several alternative ways to provide any given level of accuracy. The problem is to obtain the desired accuracy at a minimum cost. This means that the procedures used to obtain the accuracy are either combinations of several classes of procedures or a single class of

procedures. The least expensive alternative obviously depends upon the prices of the inputs, and their quantity needed to obtain the desired accuracy. In fact, the least cost alternative is given by the rule: a dollars worth of any resource should add as much to the accuracy as a dollars worth of any other resource.

The production of accurate data requires various inputs to the information system. These inputs as described earlier in the chapter can be considered as factors of production, and are the resources already available or obtainable to the organization. If the resources are already available to the organization, their quantity may not be adequate to meet the system requirements and more will be required.

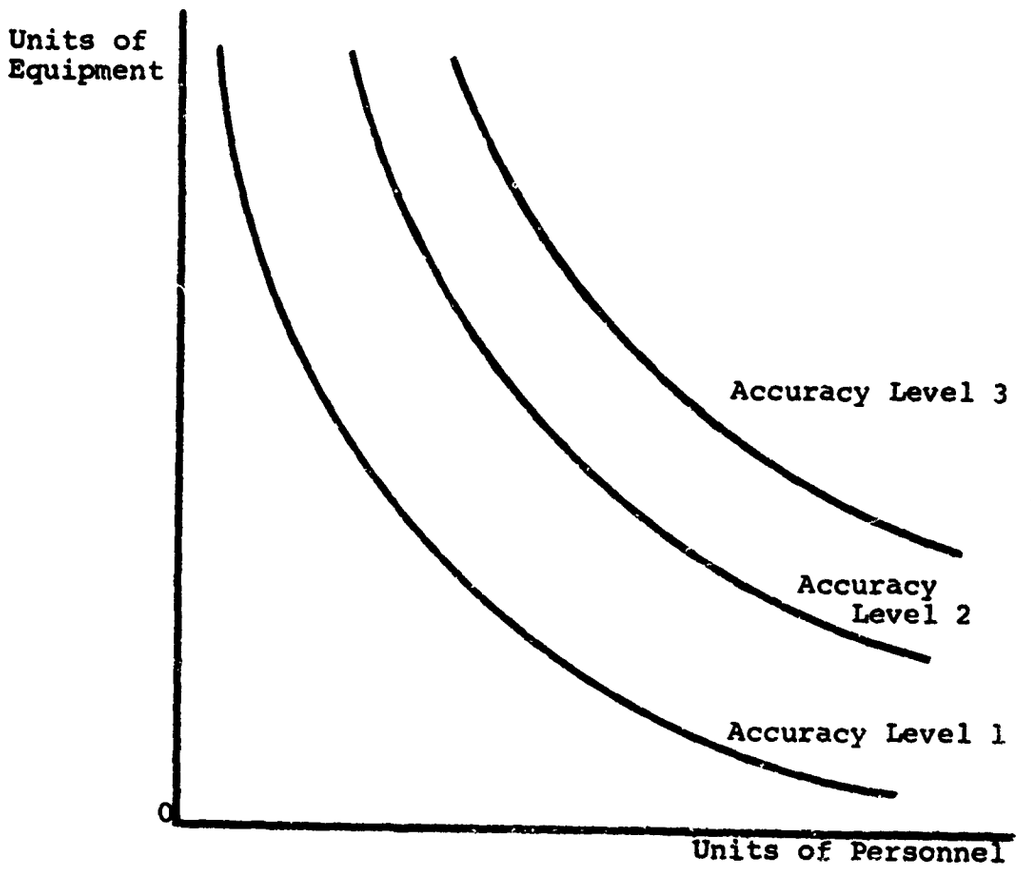
This may also be the case where some of the available resources are fixed in nature, at least for the short run, and additional units are not available. Such a constraint places a limit on the output level of accuracy that can be provided through the least cost combination of all the inputs. Independently of whether the resources are currently available or procured, their use in the detection and correction process is a cost chargeable to the procedures.

Consider two of the aggregate resources: those of personnel and computers and/or EAM equipment which were described earlier. In this case these are two basic resources associated with the detection and correction procedures of the system. Figure VII displays a set of equal accuracy curves (isoquants) which states that for each curve a given level of accuracy can be obtained using various combinations of personnel and equipment at each accuracy level. As one progresses from Curve 1 to Curve 3, the accuracy increases, but with a higher requirement for equipment and personnel.

While Figure VII gives the relationship between the units of personnel and equipment needed to obtain a given level of accuracy, there is a need to know the costs of the resources. If it is assumed that the ratio of the costs is

Figure VII

THE ACCURACY PRODUCTION FUNCTION



two personnel units to one equipment unit, then a total outlay curve (isocost) can be drawn at that ratio for the two resources. The point of tangency between the outlay curve and the desired accuracy will be the minimum cost point for the combination of resources to meet that accuracy level.[1]

Figure VIII repeats Figure VII with such an outlay curve superimposed over the isoquants.

The outlay curves of Figure VIII are parallel lines that give the ratio of prices between personnel and equipment. Line AB is tangent to Curve 2 at point 0 . Such a tangency then states that the amount P_1 , and E_1 are the units of personnel and equipment, respectively, that should be used to meet the accuracy requirements of Curve 2.

The Effect of Constraints on the Cost of Accuracy

A previous discussion describes the justification for the relationship between the resources, the accuracy levels and, by example, a method for determining the value of accurate data to the user. However, the question of how much error detection and correction can be performed is based on the cost of doing the detection and correction. The more efficient the information system and the fewer constraints placed on the system by the users, the better the system can meet the accuracy requirements at lower costs. That is, the available resources and the procedures can be integrated in a manner that is less costly when the constraints imposed by the users'are general, rather than when the constraints are specific.

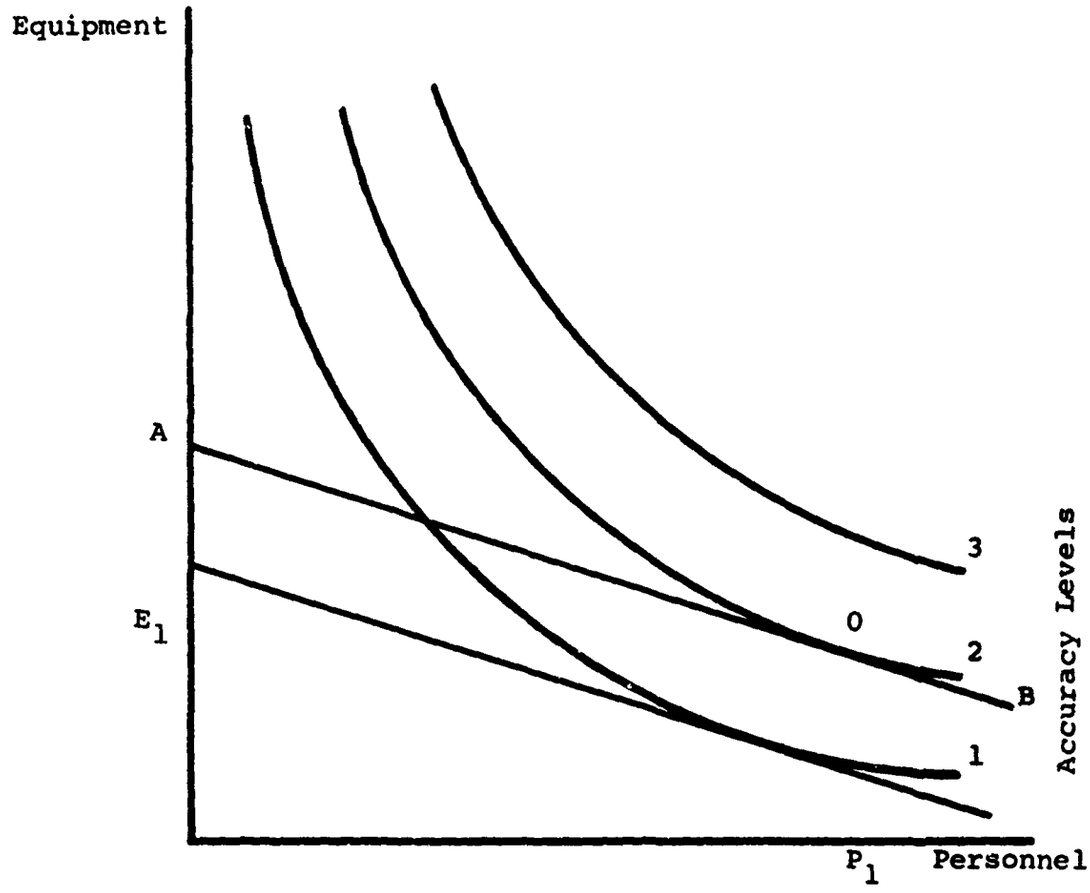
An example of such a specific constraint would be the order in which output reports are to be generated. It is quite possible that the order chosen by the users places restruction on the system manager that requires duplication of several detection and correction procedures. Such duplications increase the cost of providing a given level of

---

[1]James M. Henderson and Richard E. Quandt. Microeconomic Theory - A Mathematical Approach (New York: McGraw-Hill Book Company, 1958), p. 14. The mathematical proof is developed.

Figure VIII

OPTIMUM COMBINATION OF INPUTS



accuracy and actually cause the accuracy level to be less than optimal for the resources that are being spent.

While the need for such a specific ordering of the output may be required by the user, the system designer should make the user aware of the cost of such a requirement. From a cost-effectiveness point of view, the user can determine whether the requirement is justified in light of the additional cost.

In general, the system designer should provide the various alternatives to the system users. The users should not be expected to suggest alternatives but should become aware of the cost of the specific requirements and possible alternatives that are less costly. However, the alternatives must still meet the users' information requirements in a timely fashion.

In most cases the value of data to users changes from data element to data element as well as from user to user. This is more the rule than the exception. The system designer therefore, must decide what level of accuracy should prevail for each of the data elements. It is quite possible and reasonable to assume that some data elements are easier to bring to a given level of accuracy than other elements. This may require the system designer to perform the cost analysis at the data element level rather than the system level. If this is the case, the previous analysis will still hold for obtaining the relationships between the user, the value of the data and data accuracy.

Estimating the Cost-Output Relation

In the development of an integrated information system within an organization, some of the resources will be shared by the different components of the system. The sharing of these resources provide additional constraints on the system, especially for those resources that are costly such as the computer. If the present computer capacity is not adequate to meet desired accuracy, the costs of obtaining the additional capacity may be more per unit than

the current capacity. This is true mainly because the resource cannot be obtained in the exact kind or number of units needed for the desired accuracy. Such a case is quite possible when new procedures are incorporated into any ongoing systems.

A Currently Operating Information System

To implement additional error detection and correction procedures in a current operating system forces constraints on the kind of procedures that can be implemented. In the short run most of the resources are fixed and the system can only increase the accuracy of the data by varying the amounts of the variable resources that can be obtained. The fixed resources are in the area of equipment that has already been rented or purchased, programmed, and allocated to the various locations within the organization. Since such equipment has already been allocated, and since the system is in operation, the current availability of both computer capacity and programming talent for developing computer-aided detection and correction procedures could be quite limited.

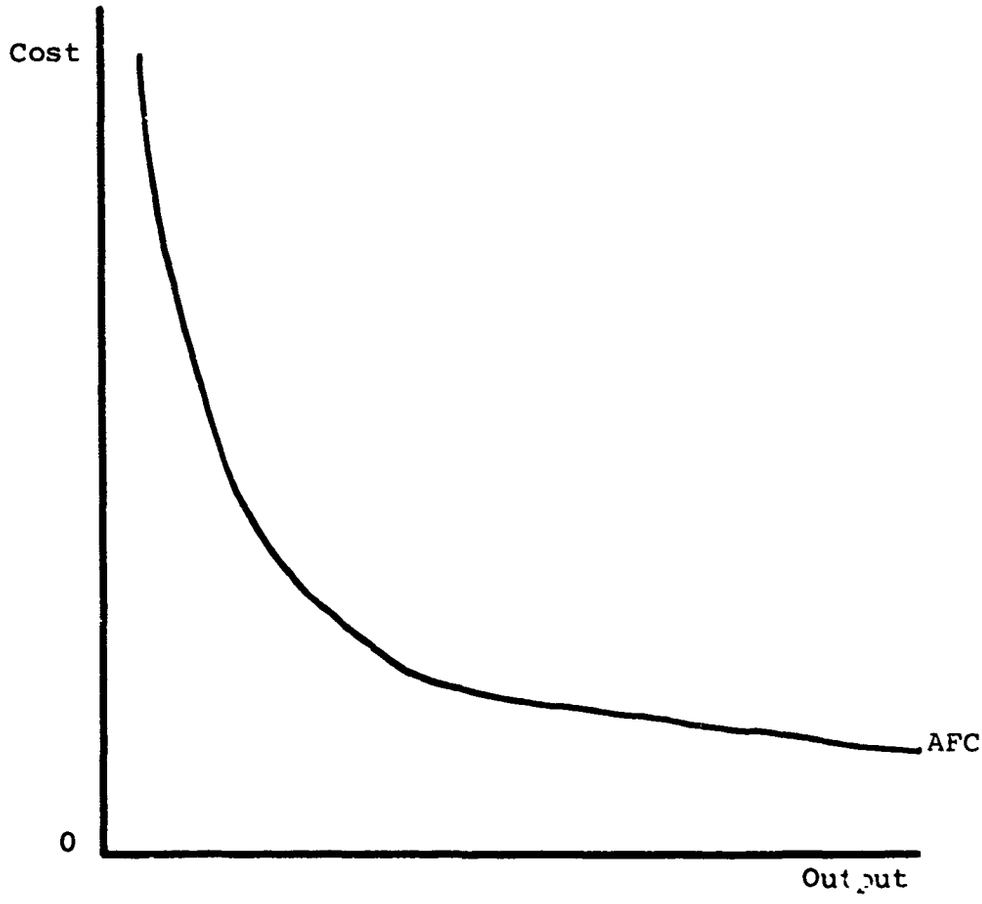
Consider the case where the computer capability throughout the information system is at capacity. However, other personnel (non-programming) can be obtained to perform error detection and correction of less complexity than the computer-aided procedures.

In this case the computer is a fixed cost to the organization. As a fixed cost, the average fixed cost (AFC) per unit of output decreases as the capacity of the computer is reached. Figure IX displays this relationship.

The computer is but one of the resources that is fixed during the short run. In addition, programmers, system analysts and data analysts are also fixed to a certain extent. This is especially true if the requirement for obtaining the resources specifies the need for knowledge concerning the current system operation, procedures and programs.

Figure IX

COMPUTER COSTS PER UNIT OF OUTPUT



Since a majority of the costs are fixed, there are fewer alternatives open to the system manager for improving the accuracy of the information system. Assuming the computer is operating at capacity, the amount of error detection and correction that can be implemented must be obtained by:

1. Some reallocation of current resources, or
2. By increasing the efficiency of the current operations, or
3. Obtaining additional units of the variable resources.

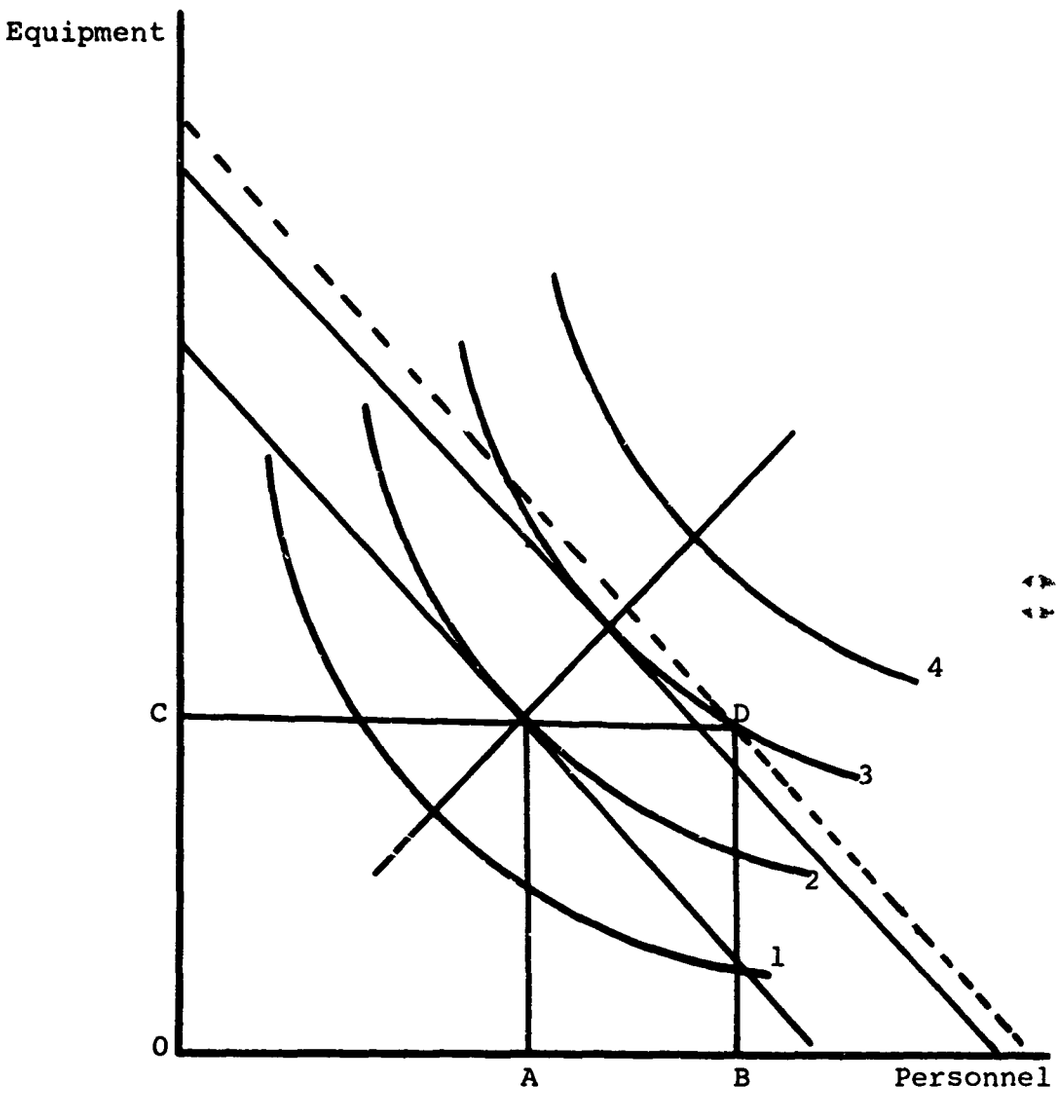
In any case, the majority of the procedures that will be implemented during the short run will be those that require very little of the fixed, already fully committed resources. Since such resources are not available to the system manager, the short term implementation must use the detection and correction procedures that are comprised of variable costs. These include system procedural changes, and the procedures associated with the data generator, the data checker, and the keypunch locations.

Figure X displays the situation where a constraint is placed on the amount of equipment that is available for use in the detection and correction process. Assume that the equipment is being operated with "perfect efficiency" and no increase in efficiency is possible. To increase the accuracy level from Curve 2 to Curve 3, additional personnel resources on the amount of AB would be required.

Since the curves of Figure X are isoquants, the cost of providing the accuracy of Curve 3 is greater than would be necessary if the proper amount of equipment resources were available. The additional cost can be measured by the difference between the total outlay curve (isocost) that is tangent to accuracy level 3 and a total outlay curve associated with point D . When this difference is known, the users must then determine if these variable costs associated with the increased accuracy are acceptable when equated to the value of data for their decision process.

Figure X

RELATIONSHIPS BETWEEN EQUIPMENT AND PERSONNEL



If the fixed equipment capability is greater than that allocated to the detection and correction process, the increased cost of obtaining the variable resources may require the system management to reevaluate their priorities concerning the allocation of equipment. The new allocation could come about if the increased costs were unacceptable to the information users, while the minimum cost of producing the required accuracy by the optimal combination was acceptable to the users.

Up to this point the equipment (mainly the computer complex and communications network) has been considered a fixed cost to the information system. However, the equipment could be considered as a variable cost to the information system, although owned by the organization. In this case the information system is but one user of the computer complex. Keeping to the short run, the system manager would use the same analysis, except the constraint on the equipment would be removed. The information system manager would purchase the additional capacity equal to the isocost line associated with the given accuracy level required. In fact, using Figure X, the system manager will expand from one accuracy level to the next along a line connecting the points of tangency of the isoquants and the isocost lines.

Such an analysis assumes that the total requirements for error detection and correction does not exceed the equipment capacity of the total organization. In addition, the price of the computer to the error detection and correction process does not increase as full capacity is reached and the resource becomes a scarce input.

The Design of a New Information System

In the previous section the cost of error detection and correction were determined by the current configuration of the information system. Most of the resources were fixed and fully committed. The system manager was required to use procedures that did not minimize the cost of the detection

and correction process. Furthermore, the depth and range of the error detection and correction were not sufficient to meet the system requirements for accuracy.

However, in the case of planning, designing and implementing detection and correction procedures into a new system, the constraints on fixed resources are removed and the resources are considered variable. That is, the new system can select the proper size facility to meet the accuracy requirement at the minimum cost. Such an analysis is different from the previous case where the system was required to operate within a set of facility constraints. In this type of analysis the designer is attempting to find that level of accuracy (output) that is minimal for all possible accuracy levels of interest to the organization.

New technology is one of the major causes for the economics of increasing returns to scale. The responsibility of evaluating the different sizes of the operation to find the most economical size for the system requirement belongs to the system designer.

Figure XI displays the relationship between required accuracy (output) cost. The long-run average cost (LAC) depicts the path for the system designer to follow in his planning. The LAC curve shows the minimum cost for any output. For this reason the LAC is often referred to as the "planning curve" of the organization.

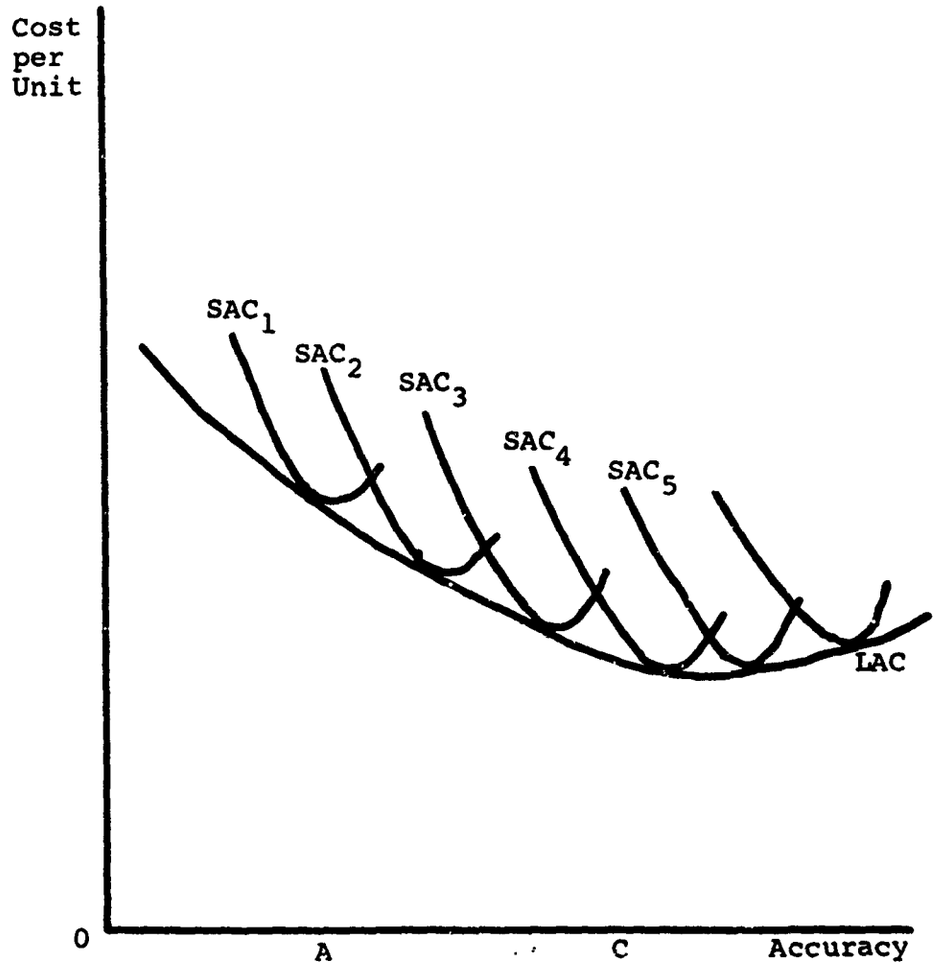
In Figure XI, the LAC is tangent to the lower portions of the short-run average cost curves (SAC) and the LAC forms an envelope of the SAC curves. Clearly, the proper system to acquire depends on the accuracy requirement (output).

For example, if the accuracy requirements can be met from a system with a short run cost reflected by SAC, then that system should be installed. However, if the accuracy requirements approach point C , then the system reflected by SAC should be installed.

The objective is to reach the system which will provide the optimal output which includes the required accuracy. That is, the capacity of the optimal system (minimum point of

Figure XI

THE PLANNING CURVE



the LAC), may be greater than the actual accuracy requirements. Yet the costs of the system reflected through the short-run costs of the optimal LAC system are less than the short-run costs for any other system that could be installed.

Summary

The chapter has described in general economic terms the costs associated with error detection and correction. Two main points were made. First, there is a need to obtain a relationship between accuracy and data worth or value from the system users. The level of the accuracy determines the amount of resources that will be required. The consumption of resources incurs costs to the system. The cost incurred by the system can be considered a fixed cost or a variable cost. Using equipment cost as an example, two situations were described. That is, the computer cost was first considered as a fixed cost, and then considered as a variable cost.

The second main point is that the detection and correction procedures can be classified by their complexity and resource requirements. In describing the various classes of procedures, three main resources were considered. The resources included the kind of personnel needed, the effect of the procedures on equipment and communications, and the data requirements of the procedures. In discussing these three aggregated resources, the attempt was made to include both quantitative as well as non-quantitative aspects of the resources. In this manner it was hoped that the cost information on the non-quantitative elements would be of some importance to the reader.

CHAPTER VII

DATA AGGREGATION FOR MANAGEMENT REPORTS

Introduction

The collection of data in itself does not produce useful information to management for decision making. To be useful, the data must be organized into arrays that have meaning to the users in their decision process. Early in the design of an information system, when the requirements for data elements to be collected are formulated, preliminary displays for the data should be developed. Important to the development of the displays is an understanding of the interaction and aggregation of the data in the decision process. This chapter will discuss some of the problems associated with data aggregation in management reports, with specific attention given to the accuracy of the data and its effect on management reports.

The Effect of Data Aggregation

In the aggregation of any data, some information is lost. The amount of the information lost can be attributed to several factors. These include the accuracy of the data themselves, the computational methods used in the aggregation process and the amount of data aggregated.

To understand how such factors affect aggregation, consider an example from the maintenance environment. Here the concern is on the mean time between failures (MTBF), as an estimate of reliability for a given piece of equipment.

The first factor affecting the aggregation is the population of the equipment. More specifically, the number

of units that failed for which the data were collected affects the aggregation. A small sample of such equipment failures may provide incorrect estimates. More seriously, several of the observations may be on the same unique piece of equipment which would suggest that the sample is biased toward one highly sporadic piece of equipment.

The second factor, computational methods, enters when the mathematical formula are applied to the observations. As the observations are divided and multiplied, additional digits are generated. These digits are dropped off the computation, thereby creating a rounding error. If each observation from the example were rounded by as much as one unit of time, the resulting solution could be quite inaccurate.

The third factor, data accuracy, enters as the basic element in the computation. If errors in the recorded data are not corrected or at least detected and marked as errors, the solution could be quite inaccurate. Consider an incorrect data entry of 100 hours for equipment operating time when the actual number was 10.0 hours. Such a large error, depending on the sample size, would bias the solution. While this example is specific in the maintenance world, the errors that can be generated by these factors are common to all information systems.

The process of eliminating the effect of data inaccuracies as well as minimizing the effects of the other factors involved in aggregation is to understand their interactive relationships. As a result, knowledge is gained concerning the direction as well as the amount of bias associated with the various aggregations.

Data Accuracy in Management Reports

In developing data displays for management there is currently very little attempt made to include some statement concerning the accuracy of the report. It is generally assumed that the report is 100 percent accurate, and in many cases does not take the decision maker or his analysts long to find a significant error.

Generally the error is due to data inaccuracy rather than an inaccuracy in computation, especially if the computations are not complex. The severity of the error will affect the future usefulness of the report to the decision maker and his decision process. The more severe the error, the less faith the decision maker has in the information presented in the report. Finally, a point is reached where the report is completely ignored by the decision maker.

Since it is impossible to eliminate all errors from the report, there is a need to communicate to the decision maker or his analysts information concerning the direction and degree of these inaccuracies. In report aggregations, the data elements will generally fall into two classes: independent and dependent elements.

Independent Data Elements

Independent data elements are defined as those elements that are context free, where the context is described as a general category of the information system. Examples include a ship name, equipment name, location, and major organization divisions, etc. These are the data elements that are considered the major characteristics on which data are aggregated. It is these data elements that provide the user with the first insight of report accuracy.

What can be done to increase this report accuracy? In general, two approaches are required. The first is a substantial error detection and correction routine to eliminate as many as possible of the errors. In general, the errors associated with these data elements can be both detected and corrected. This is possible since most independent data elements are known in advance and files can be organized to check on their accuracy.

The second method of providing for more accurate data in this class is to provide each detectable error with an error "flag". Such a "flag" would eliminate the record from any management report where that data element was a parameter of the report. In this manner detected but not corrected errors would be eliminated from the reports.

In order to inform management of the fact that not all records were used, a method of presenting such statistics should be available in the report. One such method would be to include in the report the number of records scanned, those records eliminated because of a detected error and those records that were in error, but corrected. Such statistics would give the decision maker some indication of the completeness associated with the data contained in the report. Other methods could be developed in a similar fashion to inform the decision maker of record counts or percent of coverage that is displayed in the report.

Since these data elements are independent and are, in general, the major categories by which the reports are organized, they function as control data elements and, in turn, should be controlled. The procedures used in the aggregation, therefore, require that the highest control be the most accurate, since aggregations below it will not require the same degree of accuracy. This is illustrated by the fact that the further up a hierarchical structure one moves, the less detail that is needed by that level.

Consider the following hierarchical example; the major control is the plant, of which there are four; within each plant there are fourteen departments; within each department there is equipment. This equipment consumes maintenance resources, man-hours and spare parts, both of which can be transformed into dollars. The report is to display the maintenance costs by department, by plant and by the organization. Further, assume that the data are collected against the equipment. In order to produce the report, all equipment which consumed resources are organized by department for each of the plants.

Consider the situation where an equipment code is in error, yet the department and plant are easily identified and known to be correct. The question of concern is whether the record should be eliminated since the resources consumed cannot be tied to the specific equipment. It would seem that the record is both valid and of concern to the decision

maker since it represents a cost to the organization, even if it cannot be attached to a specific equipment. Incorporating all the valid costs requires that records which have errors in specific combination be accepted by the decision rules, while other records containing errors are not accepted.

The major point of the above example is that data records known to be in error contain information that is useful and important. These records should not automatically be eliminated from the system if a uncorrectable error is contained within the record.

Dependent Data Elements

Dependent data elements differ from independent data elements in the sense that the dependent elements are usually the object of the report. That is, they are the elements that the decision maker is interested in controlling within and between the independent elements. Since the major separation or divisions are on the independent elements, the aggregation of the dependent elements becomes a function of the hierarchy of the report. While the same factors that affect the aggregation of independent elements affect the dependent elements, the aggregation of an uncorrectable error is not accepted in the same manner.

Consider the example that was described earlier concerning plants, departments and resources consumed on equipment. In that example it was suggested that records should not be eliminated from the data file if one of the independent elements of the record was in error. In the case of the dependent data elements, the element must be correct, independent of the level to which the report is aggregated. This means that errors in the dependent data elements are more important to management than the independent elements. As such, these elements should receive more importance in the error detection and correction procedures.

The error detection and correction procedures used for dependent data elements tend to be statistical in nature for both the detection and the correction. Being statistical

procedures, the detection methods allow for some variation in the recorded data elements, while the correction methods provide an expected or average value for the data element.

The Use of Uncorrected Data Elements

In aggregating data for management reports, the general theme has been to use the procedures of the accountant. That is, the report is composed of the sum of lower levels of data. If this process is used, the resulting report contains totals and sub-totals that balance. Such a procedure eliminates data and information that could be important to the decision maker, since it requires all of the data to be accurate at all the levels of the report.

Most management information systems are organized to maintain both historical and current data. The data are maintained in a record format for ease in data retrieval. That is, the entries on one record contain all the data about a specific operation. These records are then maintained in file by either a subject or time ordering. When this procedure is followed and the acco ntant's method is used, the data selected from the file for a report contain only those records that are error free. In fact, the file itself may contain only records that are 100 percent correct. The other records are discarded because of the known errors. Such a procedure does not provide management with all the information available to the organization from the data collected.

Consider the example discussed earlier concerning the four plants. Assume each record contained the independent data elements: plant, department and equipment and several dependent elements. Assume one of the dependent data elements pertained to the resources consumed by the lowest independent element which is the equipment. If each of the data elements could be considered correct or incorrect, there would be sixteen possible combinations of the four data elements, the plants, the departments, the equipment, and the resources consumed.

Table III displays the sixteen possible combinations. If the accountant's procedure was used, then only the first combination would be acceptable. There are, however, other combinations that are acceptable depending on the level of aggregation.

There are combinations that are not acceptable and these are the ones which contain an error in the resource data element. If these errors are not correctable, the combinations are not acceptable. Yet management should be told, as part of the report, something about the magnitude of these errors. When these combinations are eliminated, the remaining combinations all contain information that is important to management.

Consider a series of reports, the lowest report directed to the department heads, a second report to the plant heads, and a third report sent to the division heads, a fourth report sent to the executive office of the organization. The reports would consist of the following combinations:

For the department heads -- combinations one, four, five, and eleven; for the plant heads -- combinations one, three, five, and ten; for the division head -- combinations one, three, four, and nine. Finally for the executive officer of the organization, combination fifteen would be added to the other seven of the three other levels.

To implement such an aggregation procedure requires that the detection and correction methods include a means of indicating when a detected uncorrected error exists in the data file. With this knowledge report, generation programs can be written which will incorporate the maximum amount of data in the report.

Data Precision in Uncorrected Data

In the previous section an aggregation procedure was described which used correct data that were contained in a record where some of the required independent data elements were in error. The intent was to show that data aggregation may depend on a record concept to obtain and describe the

Table III

TRUE-FALSE TABLE FOR DATA AGGREGATIONS

	Plants	Depart-ments	Equip-ments	Resource
1.	T	T	T	T
2.	T	T	T	F
3.	T	T	F	T
4.	T	F	T	T
5.	F	T	T	T
6.	T	T	F	F
7.	T	F	T	F
8.	F	T	T	F
9.	T	F	F	T
10.	F	T	F	T
11.	F	F	T	T
12.	T	F	T	F
13.	F	F	T	F
14.	F	T	F	F
15.	F	F	F	T
16.	F	F	F	F

information. However, all of the data elements of the record need not be error free for each level of the report.

The procedure increases the information made available to the decision maker, as well as the utilization of the correct data in the data bank. Nevertheless, there remains a question concerning the individual recorded data elements that are uncorrectable. In the case of the independent data elements, after all correction procedures have been exhausted, there is little that can be accomplished. For those cases it would be best to discard the uncorrected data element. In discarding the data element from the record, a code should be placed in that data field of the record. The code is used to indicate that the element recorded was in error. It could not be corrected and was discarded.

In the case of dependent data elements, the problem is of a different nature. Here the statistical methods used for detection and correction can have a wide variation in what can be considered acceptable to the data base. Therefore, the narrower the limits placed on the detection process, the fewer the number of extreme values that will be admitted. In turn, these extreme values will be considered in error and, where possible, will be corrected to either an expected value or an average.

In determining the range of values that will be accepted through the statistical procedures, the limits of the range may be too large for the required aggregation. Consider a data element that, through historical data and a majority of the users, accepts a range of plus and minus three standard deviations. Assuming normality, the range would contain 99 percent of the observations if the true mean were, in fact, equal to the historical mean used in developing the range. For particular aggregations, specific decision makers may require only two standard deviations, or at least some number less than three standard deviations.

The development of ranges and limits is required since the dependent data elements are not corrected by an exact procedure. In these cases there is a need for an

indicator or file of data descriptions and declarations to be maintained separately within the system. In this manner the necessary information concerning the requirements for data correction via a detection logic would be available. The statistical data needed to change the limits on the data element now accepted as correct would be included. Such a separate file would allow the decision maker to request data at any level of precision he desired, remembering that the precision is based on a statistical average and variance.

For dependent data elements that are uncorrectable, there is little value in retaining them for use in management reports. Such uncorrectable errors should be separated from the basic data file and used by the data system analysts to develop new detection and correction methods.

Report Reliability

Earlier discussions alluded to a method of providing an estimate of report reliability to the user. Such a method suggested the use of ratios of records or data elements that were in error to the total of the records scanned to make the report. The main objective is, however, to convey to the user a feeling of security or accuracy in using the report in his decision process.

The procedure by which such a method is presented to the user depends on the kind of report being received by the user. There are, in general, three kinds of reports that will be generated by the system: 1) the periodical report, 2) the time series or cumulative report, and 3) the exception report. A fourth kind of report which would follow one of the three basic formats is an on-request report, which is a special case of the exception report. That is, the on-request report is initiated by the user, while the exception report is generated by actions that occur within the information system.

The Periodical Report

The periodical report is based on a common set of data elements and a fixed time period. Since the same data elements are presented in each report, the report user can quickly become familiar with the meaning and use of reliability statistics. The statistics would provide the kinds of data that enable the decision maker to evaluate the appropriateness of the report to his decision process. The statistics would include such data as:

(1) The beginning and ending dates of the report.

(2) The coverage of the data -- a measure of the population that the report tends to include.

(3) The depth of the coverage -- a measure of the variation between the objectives in the population.

(4) Individual data element reliability -- a measure of the accuracy of each data element contained in the report. A simple procedure would be total number of observations less error observations rejected divided by total number of observations.

(5) Total report reliability -- an overall measure of the accuracy of the report as a function of coverage and data element reliability. This measure could be as simple as the product of the individual data element reliabilities and a weighted average of the coverage of the report. Other more complex methods could be developed using such relationships as independent and dependent data element joint error distributions derived from the possible data error combinations that were suggested by Table III.

The Time Series on Cumulative Reports

The aggregation of data for these reports should receive the same attention given report statistics in the periodical reports. In addition, there is a facet of cumulative reports which tends to alarm the users, yet it is the main objective for obtaining cumulative reports.

In some classes of periodical reports, attention is given to "the ten high" trouble items. In a month to month report there is likely a shifting of the items that are included in such a "top ten" list. Decision makers tend to look at each month separately, especially when the next monthly report shows that the top critical item of the previous month has dropped to a lower place in the list. However, a cumulative report tends to focus on the items that are always near the top of the list month after month. In this manner the decision maker can see much more clearly wnere his energies should be placed with respect to continuous problem areas illuminated by this kind of report.

Exception Reports

Exception reports are by nature cause and effect situations, and they require careful computational procedures for describing an exceptional condition. The report is only generated when certain limits or bounds are exceeded by the program decision rules. Therefore, specific attention to the error detection and correction procedures used in data aggregation and internal program decision rules are important. If the decision logic is complex, in that several variables are interrelated and one or more are able to cause an exception report, data statistics are mandatory. Such statistics will be required by the decision maker in locating the particular variables that caused the exception.

In order to analyze the exception and reach a course of action, the decision maker will require additional information. While the basic information will be contained in the body of the exception report, the data statistics concerning the report will provide additional information. Finally the decision maker has the option of returning to the information system for a more detailed report concerning the object of the exception report.

In obtaining additional reports from the information system, the volume of such reports may be too great for serious analysis. If this is the case, the data statistics

contained in the exception report provide a means by which the decision maker can select a subset of detailed reports available to him. To provide this insight, the data statistics needed include those described under the periodic reports. In addition, some statistics such as the range and variance of the dependent data elements should be provided. Specific bounds and limits of the decision logic that were exceeded and some measure of the degree to which the limits were exceeded should also be included. With these additional data the decision maker can intelligently select the proper subset of the reports needed to reach a decision.

Error Probabilities

In the generation of each report, the level of aggregation along with the individual data element reliabilities provide a loss of information to the information system user. Each data element has an error distribution associated with it. The error distribution is composed of two parts. The known part of the distribution is described by the errors that are detected through the error detection procedures. The unknown part of the error distribution is described by the undetected errors that are not detected by the procedures. This can be defined as the undetected error rate of the data element.

The known error rate can be separated into two parts: that part detected and corrected and the part detected but not corrected. As a measure of the effectiveness of the error detection and correction procedures, ratios between the various subsets of the error distribution can be performed and maintained.

The ratio of detected errors to total data validated will provide a basic indication of data element reliability. The ratio of corrected errors to detected errors will give a measure of correction effectiveness. Monitoring of these ratios over time will provide a continuous means of evaluating system accuracy performance.

The part of the error distribution associated with the undetected error rate is harder to estimate. Since this part of the distribution belongs to the errors that have not been detected, there is no known measure of the error.

There is, however, a method of estimating the magnitude of the true data element error rate. The method is based on the errors that are detected and an estimate of the undetected errors through the code set density factor. Consider the relationship between the data element code set and the total possible combinations of codes that could be generated by that length of code and characters composing each position of the code. That is:

$$\frac{\text{Code set}}{\text{Total possible number of codes}} .$$

This ratio defined as the code set density factor gives two insights into the detection process.[1] The first insight into the detection process is concerned with the density of the code set; as the ratio approaches one, the code set becomes more dense. The more dense the code set, the harder it is to detect erroneous codes by inspection alone, thereby requiring more elaborate detection procedures.

The second insight is related to the first in that under some general assumptions, the density ratio can be used to estimate the "true" error rate of the data element. This is accomplished by subtracting the code set density factor from unity. The result, defined as the coefficient of detectability becomes a indicator of the complexity needed in performing the detection process, as well as a basic component in determining the "true" error rate.

In the simplest case the "true" error rate can be considered as the percent of errors that are not detected by inspection alone. However, such an error rate can be

---

[1] Owsowitz and Sweetland, Factors Affecting Errors, p. 26. The authors refer to this ratio as an error-discernability factor.

expanded to include more of the detection procedures than inspection. As an example, consider the case where the data element is a two-digit numeric code. The code set is only 20 codes of the possible 100 codes that could be generated by all combinations of the numerics. In addition, assume that 500 detected errors were observed by the inspection procedures, where 10,000 records containing the data element were processed.

The coefficient of detectability is calculated as 1 - (20/100) = .8 ; while the ratio of detected error elements to total elements processed is 500/10,000 which is five percent. The error means that the data generators recorded a code they thought was a member of the code set five percent of the time.

Since the coefficient of detectability is 80 percent, there are additional errors of transposition which resulted in one code being transferred into another code which is undetectable by inspection alone. Therefore, the "true" error rate for this case must be above the five percent detected, and can be estimated by:

$$\left[\frac{\text{Detected errors}}{\text{Total records processed}}\right] \cdot \left[\frac{1}{\text{Coefficient of detectability}}\right] .$$

In this example,

$$[.05] \cdot [\tfrac{1}{.8}] = .0625 \quad \text{or 6.25 percent}$$

as an estimated "true" error rate for this data element when the inspection procedures alone are used.

It should be pointed out that the calculated "true" error rate as described in the example assumes that the error rate for the undetected errors is proportional to that of the detected errors. It seems reasonable to make this assumption if the records processed are representative of all situations present in the system.

Earlier it was stated that the "true" error rate could be expanded from one of detection by inspection to more elaborate procedures. This can be accomplished in the same manner as described above, where the number of errors detected would be a value in the procedure for estimating the "true" error rate. If each procedure was used in sequence, and each could be considered as independent, then the "true" error rate would be the sum of the individual ratios for each function. That is:

$$\text{True error rate} = \left[\frac{\text{Detected errors of Procedure A}}{\text{Total records processed}}\right.$$

$$+ \frac{\text{Detected errors of Procedure B}}{\text{Total records processed by Procedure B}} + \cdots$$

$$\left. + \frac{\text{Detected errors of Procedure N}}{\text{Total records processed by Procedure N}}\right]$$

$$\cdot \left[\frac{1}{\text{Coefficient of detectability}}\right]$$

or

$$\text{True error rate}$$

$$= \left[\sum_{i=1}^{N} \frac{\text{Detected errors by Procedure } i}{\text{Total records processed by Procedure } i}\right]$$

$$\cdot \left[\frac{1}{\text{Coefficient of detectability}}\right] .$$

The coefficient of detectability is assumed to be constant for all the procedures. The assumption may be conservative in view of the complex computer-aided detection procedures since the coefficient of detectability is a function of code set density alone. However, it seems reasonable to assume that the more dense a code set is, the more complex the procedures needed to detect errors, and the higher the possibility of erroneous data not being detected

The above procedure described a method for estimating the "true" error rate of a data element as far as detection is concerned. However, it is necessary to define a method for presenting the detected errors that have been corrected. One method that is easily obtainable from the data presented is the ratio of errors not corrected to errors detected. Another method would be to subtract from the number of errors detected the number corrected and use the resulting ratio as a measure of error probability for each of the procedures. Either method would produce the same result and would be an estimate of the actual error rate as opposed to the "true" error rate. The difference between the two rates would be a measure of correction procedure effectiveness.

Summary

The chapter attempted to describe several important problems concerning the aggregation of data into management information reports. The problem of presenting data statistics for use by the information users was described, including the identification of relationships between independent and dependent data elements in a report. The kinds of reports generated by an information system were described as was the use of data records that contain errors in aggregating management reports.

In presenting data error statistics, several methods were described for estimating the data element error rate. These methods were based on the assumption that the error rate could be described in two ways.

The first method described the error rate in terms of the detection procedures, and reflected the errors that entered the system. This was divided into two parts, the known error rate derived from ratios of detected errors to total number of data elements processed. The unknown rate was derived from the coefficient of detectability and the known error rate percentage.

The second approach was to define the error rate as the rate at which data left the system. That is, to consider how well it was corrected by the system procedures and to question what was presented in the reports to the system users. Again, the methods discussed were similar to those for detection. However, the number of errors detected was replaced by either a percentage reflecting the corrected portion, or as a difference between what was detected and corrected. For both methods it was shown that the final error rate could take account of different detection or correction procedures to obtain an overall error rate for the data element.

The two error rates are defined as the "true" error rate of the system. The first is a description of the error rate of the data entering the system and it is related to the detection procedures. The second is the error rate of the data leaving the system after the correction procedures have been applied. Each rate has a useful purpose, both in data aggregation and in system monitoring. The system manager would be wise to provide and maintain such statistics as one of the measures of system effectiveness.

CHAPTER VIII

MODELS AND PROCEDURES FOR ERROR DETECTION AND CORRECTION IMPROVEMENT

Introduction

The objective of this chapter is to provide the system designer with a means for determining the location, the range, and the depth of error detection and correction procedures. The criteria for determining the various procedures and locations will be in accordance with the status of system environment.

The Decision Array

In deciding the degree of error detection and correction needed for any particular information system, many alternative strategies must be evaluated by the system designer. More specifically, the data accuracy requirements must be tailored to the various decision processes being used as well as to the effect that these decisions have on the operation of the organization. In particular, a decision will depend on the analysis, interpretation and evaluation of the information available to the decision maker. A significant part of that information comes to the decision maker through the data generated by the information systems of the organization.

The usefulness of the information presented to the decision maker depends to a great extent on the accuracy of the data that produce the information. The total success of the information system is tied to assessing the proper level of data accuracy. If this is the case, then before decisions are made concerning error procedures, it is

important to account for all activities that interface with the error procedures. Because the system designer does not have perfect knowledge concerning the final use of the information, the decisions that are to be made contain some risk. To reduce the number of decisions which must be made under risk, the system designer is required to obtain as much knowledge as possible about the environment in which the system will operate.

One method of presenting the alternatives in an orderly manner is through a decision array. The decision array associated with the detection and correction procedures is organized around variations of the range and the depth of the procedures as well as the locations at which the procedures can be performed. These variations become the actions to be evaluated agains the different states of system environment. The end result is to determine the cost of performing the error detection and correction function.

Figure XII displays the complete set of alternatives available to the system designer. In Figure XII each cell of the three dimensional array denotes a particular range and depth of error detection and correction that can be accomplished at a particular location. The range of the detection and correction procedures is defined as the various data element classes contained in the system. The depth is defined as the various classes of procedures that can be used at the various detection and correction locations.

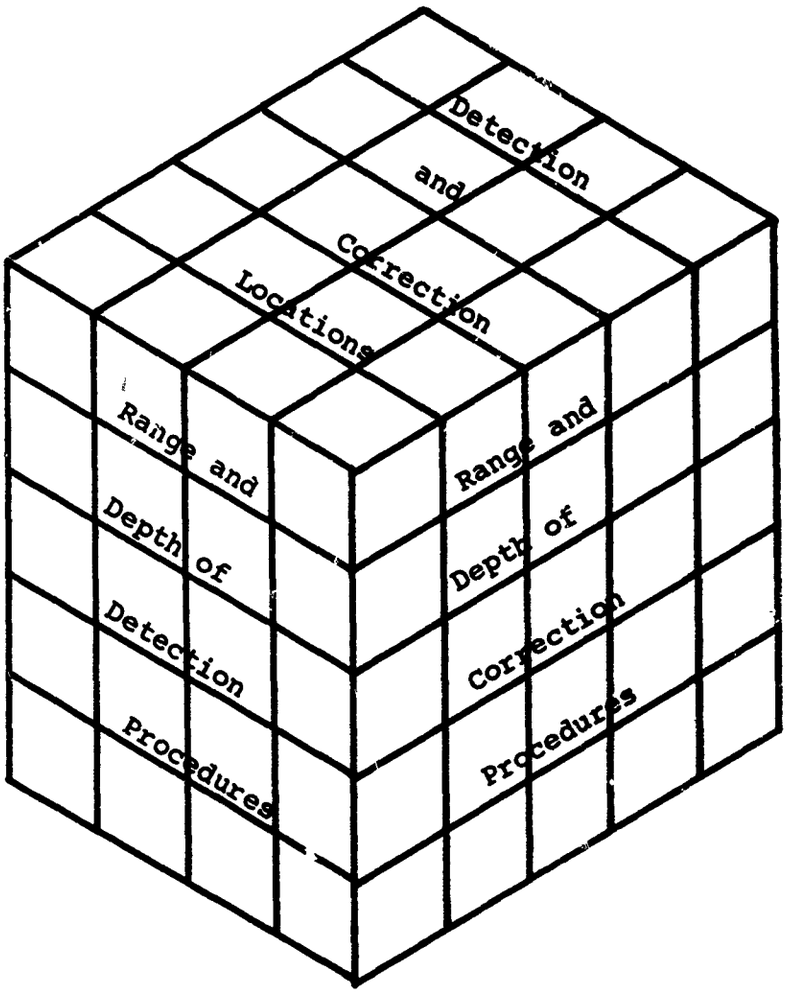
The Decision Model

Assume that there are m locations, n variations of the range and depth of the detection procedures, and p variations of the range and depth of the correction procedures, then,

A_{ijk} = the ith detection and correction location composed of the jth variation of range and depth of detection and the kth variation of the range and depth of the correction procedures.

Figure XII

DECISION ARRAY



In addition define:

R = the range of detection, where $r_1, r_2, \ldots, r_R$ are the data element classes of the system.

S = the depth of detection, where $s_1, s_2, \ldots, s_S$ are the detection procedures, and

T = the range of correction, where $t_1, t_2, \ldots, t_T$ are the data element classes of the system.

U = the depth of correction, where $u_1, u_2, \ldots, u_U$ are the correction procedures.

Furthermore, let the set $D = \{d_{rs} | \begin{smallmatrix} r = 1 \text{ to } R \\ s = 1 \text{ to } S \end{smallmatrix}\}$ be the data element classes and their associated detection procedures, and, let the set $C = \{c_{tu} | \begin{smallmatrix} t = 1 \text{ to } T \\ u = 1 \text{ to } U \end{smallmatrix}\}$ be the data element classes and their associated detection procedures.

Then n , the variations of the range and depth of the detection procedures = 2^{RS} subsets of the set D , similarly, p , the variations of the range and depth of the correction procedures = 2^{TU} subsets of the set C .

The decision array as defined has m locations, $n = 2^{RS}$ variations of the range and depth of detection and $p = 2^{TU}$ variations of the range and depth of correction. Since the number of variations in both the range and depth of the detection and correction procedures increases by the power of 2^{Z+1} for every new data element class or procedure included, care should be used in their selection.

For example, consider the following data classes and detection and correction procedures.

R = {r_1 = dynamic-factual data class; r_2 = dynamic-judgmental class; r_3 = static-factual class; r_4 = static-judgmental class}

S = {s_1 = non-computer-aided detection procedures; s_2 = computer-aided detection procedures}

$T = \{t_1$ = dynamic-factual data class; t_2 = dynamic-judgmental class; t_3 = static-factual class; t_4 = static-judgmental class$\}$

$U = \{u_1$ = non-computer-aided correction procedures; u_2 = computer-aided exact procedures; u_3 = computer-aided approximate procedures$\}$

Then

$D = \{d_{11}$ = dynamic-factual class and non-computer-aided detection procedures; d_{12} = dynamic-factual class and computer-aided detection procedures; ... d_{rs} ... d_{42} = static-judgmental class and computer-aided detection procedures$\}$

where

n = all the subsets of the set $D = 2^8$ possible subsets or 256 variations of the range and depth of error detection.

For the set

$C = \{c_{11}$ = dynamic-factual and non-computer-aided correction procedures; c_{12} = dynamic-factual computer-aided exact procedures; ... c_{tu} ... c_{43} = static-judgmental and computer-aided approximate procedures$\}$

where

p = all the subsets of the set $C = 2^{12}$ possible subsets of 4,096 variations of the range and depth of error correction.

From the preceding example, it is obvious that the number of cells in the array can be very large. In fact, the example described only two dimensions of the array and the number of cells were $256 \times 4,096 = 1,048,576$. The third dimension, the locations, would increase the number of cells to a still larger number.

To develop a decision array for all the data classes and detection and correction procedures described in the paper would be an impossible task. The tremendous size of the decision array does indicate the magnitude of the detection and correction problem facing the system designer.

However, the various data element classes and the detection and correction procedures can be aggregated to a higher level, and then "de-aggregated" as the number of alternatives are decreased. For example, using only the two major data classes, static and dynamic, and two major detection and correction procedures, non-computer-aided and computer-aided, produced sixteen variations of the range and the depth of detection and correction. Where,

$$R = \{r_1 = \text{static}; \; r_2 = \text{dynamic}\}$$

$$S = \{s_1 = \text{non-computer-aided}; \; s_2 = \text{computer-aided}\}$$

$$T = \{t_1 = \text{static}; \; t_2 = \text{dynamic}\}$$

$$U = \{u_1 = \text{non-computer-aided}; \; u_2 = \text{computer-aided}\}$$

and

$$D = \{d_{11}, d_{12}, d_{21}, d_{22}\}$$

$$C = \{c_{11}, c_{12}, c_{21}, c_{22}\} .$$

Where

$n = 2^4 = 16$ variations for the range and depth of detection procedures, and

$p = 2^4 = 16$ variations for the range and depth of correction procedures.

Table IV displays the variations for either the detection procedures or the correction procedures since both are the same in this example. Included in Table IV are the null set and the universal set. The null set is the empty set which means that none of the data element classes are subjected to any detection or correction. The universal set means that all the data classes are subjected to all the procedures.

Table IV

THE SUBSETS OF THE SETS D AND C

1. Null set.
2. Static non-computer-aided.
3. Static computer-aided.
4. Dynamic non-computer-aided.
5. Dynamic computer-aided.
6. Static non-computer-aided and static computer-aided.
7. Static non-computer-aided and dynamic non-computer-aided.
8. Static non-computer-aided and dynamic computer-aided.
9. Static computer-aided and dynamic non-computer-aided.
10. Static computer-aided and dynamic computer-aided.
11. Dynamic non-computer-aided and dynamic computer-aided.
12. Static non-computer-aided and static computer-aided and dynamic non-computer-aided.
13. Static non-computer-aided and static computer-aided and dynamic computer-aided.
14. Static computer-aided and dynamic non-computer-aided and dynamic computer-aided.
15. Static non-computer-aided and dynamic non-computer-aided and dynamic computer-aided.
16. Universal set.

In further discussions, Table IV will be referred to using the notation of set theory. In particular, where D and C are partitioned into unit sets, that is:

$\{d_{11}\} = \{c_{11}\}$ = static non-computer-aided

$\{d_{12}\} = \{c_{12}\}$ = static computer-aided

$\{d_{21}\} = \{c_{21}\}$ = dynamic non-computer-aided

$\{d_{22}\} = \{c_{22}\}$ = dynamic computer-aided .

Then from Table IV for j=6 we have $\{d_{11}\} \cup \{d_{12}\} = \{d_{11}, d_{12}\}$ while for j=14 , $\{d_{12}\} \cup \{d_{21}\} \cup \{d_{22}\} = \{d_{12}, d_{21}, d_{22}\}$. The use of the notation will minimize the repetition and present a clearer understanding of the model to the reader.

Figure XIII displays the ith detection and correction location as a two-dimension array. The rows and columns of the array reflect the variations of range and depth of the detection and correction procedures displayed in Table IV. For example, the intersection of the third row and the fourth column; j=3 , k=4 is where computer-aided detection procedures are applied to static data elements. In addition, non-computer-aided correction procedures are applied against the dynamic data class.

Consider the following examples of cells contained in an array similar to Figure XIII.

(1) A cell located at the data generator location where the range of detection is over all the dynamic and static data elements of the system. The depth of detection will include the non-computer-aided procedures of: visual scan, internal system procedures, debriefing with co-worker, short check lists, and system manuals (j=7) . The range of the correction procedures is also over all the dynamic and static data elements at that location. The depth of the correction includes the non-computer-aided procedures associated with system manuals, code books, memory recall, debriefing, and short look-up tables (k=7) .

Figure XIII

THE ith DETECTION AND CORRECTION LOCATION ARRAY

j \ k	1	2	3	4	.	.	$k=p=2^{TU}$
1	1,1	1,2	1,3				1,p
2	2,1	2,2	2,3				2,p
3			3,3	3,4			
4			4,3	4,4			
.							
.							
.							
$j=n=2^{RS}$							n,p

(2) The cell is located at the keypunch location and, in particular, the data checker section of the location. The range of detection would include both the static and dynamic data elements. The depth of detection would include the non-computer-aided procedures of visual scan, templates, check lists, bound and limit values (j=7) . The range of correction would include only the static data elements. The dynamic elements found in error would be marked and processed at a higher correction location or returned to the data generator, depending on the physical location of the generator with respect to the data checker. The depth of correction on the static data elements would use such procedures as code books, catalogs, system reference manuals and tailored look-up tables (k=2) .

(3) The cell is located at the local computing facility, where the range of detection is over the dynamic elements and the detection procedures include computer-aided admissibility checks and simple joint code set relationships (j=5) . The range of correction is over the static elements and the depth of correction includes the computer-aided procedures of error correcting digits schemes, joint code set relationships and statistical limits (k=3) .

Model Operation

The system designer is faced with the problem of eliminating from the array those cells that are not responsive to the particular information system under consideration. The intent of the system designer is to reach a manageable number of alternatives that can be evaluated as to cost, efficiency and resulting accuracy.

In order for the system designer to reach this objective, he must be aware of the various states of system environment. That is, the system designer must determine the range and depth of the detection and correction procedures by evaluating the environment in which the system will be operating. Because of the size and complexity associated

with any large information system, the designer must use a systematic approach. The decision array described earlier is intended to be the model from which the designer evaluates and selects the best combination of locations and procedures.

The model functions as an elimination process through a set of procedures that are iterated for each new alternative. The procedures describe the bounds of the alternatives and proceed to eliminate alternatives as more knowledge is gained concerning the states of system environment. The procedures are separated into two parts. The first set of procedures eliminates alternatives from an aggregated decision array; the second set of procedures separates the remaining aggregated cells of the decision array into finer variations of the detection and correction procedures. The point of transfer from the first set of procedures to the second is subjective and depends on the system designer's judgment as to the number of alternatives which can be easily managed.

The model operates in the following manner.

1. Define the initial bounds on the size of the decision array.

   (a) Determine from the system definition/objectives the data element classes for the system. The classes define the elements of the sets R and T. These sets will be used to determine the variations in the range and depth of the detection and correction procedures.

   (b) Determine from the system objectives the detection and correction procedures available to the system. These procedures define the set S for detection procedures and the set U for correction procedures. These sets will be used in determining the variations in the range and depth of the detection and correction procedures.

   (c) Determine the number of detection and correction locations available within the system. The number of locations will vary between two, for an on-line real-time system, to seven or more for a general batch processing system.

(d) Formulate the sets D and C . These sets determine the variations of the range and depth of the detection and correction procedures.

(e) Prepare a two-dimensicn array displaying the subsets derived from the sets D and C . This two-dimensional array will be the same for each location, and will be similar to that of Figure XIII. The array will be used as a model for each location and as a guide to eliminating alternatives from the location.

At this point the decision array is defined and includes all the alternatives open to the system designer, as defined by the data classes and procedures.

2. Using the two-dimension array prepared above, determine, for each location, the data classes that are not available at the location. Eliminate all variations of the range and depth of the detection and correction procedures that include those data classes.

3. Using the modified array resulting from (2) above, determine for each location the classes of procedures that are not available at the location. Eliminate all variations of the range and depth of the detection and correction procedures that include those classes of procedures.

4. The elimination process of steps (3) and (4) reduce the number of rows and columns for each location. The rows represent the elimination of variations in the detection procedures, while the columns represent the elimination of variations in the correction procedures.

5. Correction procedures cannot be employed to data classes until errors have been detected in the data classes. The constraint requires that detection procedures precede correction procedures. In the elimination of rows and columns in steps (2) and (3), inconsistencies may have resulted. These inconsistencies are corrected by eliminating the particular cell causing the inconsistency. The detection need not be performed at the location where correction is anticipated. However, the detection must be performed at some location previous to the correction.

6. After any inconsistencies are corrected, the resulting two-dimensional arrays represent the alternatives available to the system designer after the first iteration.

7. The second and future iterations depend on the knowledge available to the system designer which is obtained from the questions concerning the state of system environment. That is, starting with step (2) above, the system designer evaluates the remaining variations at each location. However, starting with the second iteration, the emphasis is on the elimination of particular cells of the array rather than the complete elimination of a row or column. Yet both can be eliminated during any iteration when an unacceptable alternative is discovered.

8. When the alternatives for a location reach a point where further elimination is not possible due to the level of aggregation of the variations, a more detailed decision array is possible. That is, within the remaining variations, identification of particular sub-classes of data elements and particular detection and correction procedures can replace the aggregated alternatives.

   For example, consider the case where the data element classes were originally defined as either static or dynamic. If the final alternative at a location is concerned with only the static data class, a more detailed evaluation could be made by separating static into factual and judgmental. At the same time the specific detection and correction procedures for these more detailed data classes would be identified.

   To move from a higher level of aggregation to a lower level within the model assumes that consistency between the levels will be maintained. That is, the new data classes and procedures are part of a hierarchical scheme contained within the higher levels originally used in the model.

   For any location the procedure for defining the new decision array is the same as described in step (1) above. After the dimensions of the array for each location are defined, the remaining steps of the procedure described previously are used. Through

the iteration process the system designer will reach a subset of the cells that are best for each location.

The remaining alternatives, which cannot be further evaluated from the system environment view, must be evaluated from a cost perspective. That is, all the alternatives are equally acceptable from a system feasibility view, but the cost of performing the alternatives are not equal.

While the first cost estimates may be only subjective estimates that provide an ordinal ranking of the alternatives, several of the less attractive alternatives will be eliminated and further refinements in the cost estimates of the remaining alternatives will be required. In general, the cost estimates will become available as the preliminary system feasibility develops and an iteration process takes place between the overall system design, the states of system environment and the accuracy requirements of the system.

The states of system environment describe for the system designer the constraints placed on the system. These constraints affect not only the error detection and correction function, but the total information system organization and operation. In order to evaluate the effect of these constraints and system operations on the error detection and correction function, a check list of questions which, when answered, will determine the state of system environment.

The Check List

The check list will provide the necessary information to determine the location, range and depth of the error detection and correction procedures. While far from complete, the check list is intended to provide some of the necessary key questions that need to be answered before an intelligent decision can be made concerning the range, depth and location of the error detection and correction function.

The check list is composed of the major areas of the information system that relate to the error detection and

correction function. The major areas selected for this check list are (1) the organizational environment of the system, (2) the goals and objectives of the system, (3) the system hardware environment, (4) the data collection and data availability environment, and (5) the system user environment.

The Organizational Environment of the System

The organizational structure, as a state of system environment, is defined to include those aspects of management which plan, direct, control, organize and staff the organization in which the system is to operate. A sample of the important questions would contain:

1. Is the parent organization centralized or decentralized?
2. What is the extent of centralization or decentralization?
   (a) Functional areas
   (b) Geographic areas
   (c) Complete central authority
3. What is the geographic spread of the organization?
4. Is the information to support all levels of management?
   (a) Does the lowest level of management include day-to-day operations?
   (b) Does the system act as a process control system as well as an information system?

(c) How many levels of management are there in the organization?

(d) What is the degree of decision making power at each of the management levels?

5. Is the system to support all the functions of the organization?

(a) How are the functions that are to be supported, dispersed throughout the organization?

(b) Is this the only information system supporting these functions?

6. Are there any anticipated organizational changes that will affect the system?

(a) Will the location of detection and correction procedures change?

(b) Will the range and depth of the procedures change?

(c) Will the resources required for detection and correction change?

System Objectives and Goals

The system objectives and goals affect the error detection and correction function through system operation. The following sample questions reflect the goals and purpose of the system.

1. Is this system development for a new system or a modification of an existing system?

(a) What prompted the requirement?

(b) Who is the sponsor of the system and what are his interests?

(c) Has a limit been placed on the resources for system operation?

2. Determine the precise purpose of the system for

(a) The functional areas to be incorporated;

(b) The expected improvement to be obtained.

3. Determine the relationships with other systems of the organization as to:

   (a) Authority concerning data sharing and data validation;

   (b) Schedules of interfaces between inputs and outputs.

4. Determine the relationships between the system and systems outside the organization:

   (a) Which systems feed information (data) to this system and to which system does it provide output (data/information)?

   (b) What procedures are now in existence between the organization and the outside systems? How will these procedures be affected by the new system, including the method of data transfer, period of transfer, machine compatibility.

5. Do the goals define the management functions that are to be supported?

   (a) Have the goals been approved by management?

   (b) What future growth in the functions is envisioned?

6. What is the expected life of the system?

7. What degree of system flexibility is required?

8. Is the system to be designed through modular subsystems?

9. Is there a time-phased plan for implementation?

10. What is the system response time?

    (a) Are there specified response times for different actions?

    (b) Are there specified response times for different management levels?

11. What security measures are required, and how have they been formulated?

12. What is the requirement for system feedback to lower management levels?

13. What is the requirement for system performance indices?

14. What are the requirements for maintaining data permanency?

15. What reliability has been placed on system operations?

16. What backup system is anticipated?

17. Determine what standards are necessary for uniformity in subsystems, procedures, formatting, and system documentation.

System Hardware Environment

For the error detection and correction procedure, the system hardware configuration, its location and mode of operation are of prime importance. In addition, the final selection of the specific hardware will determine the exact programming techniques that will be used to develop the computer-aided programs.

1. What is the hardware system in functional elements?

2. Where will the hardware be located?

3. Describe the functional elements that will be available at each location.

4. Is the hardware currently within the organizational structure?

5. Will new hardware be obtained?

6. What are the current computerized functions being performed by the organization?

7. What will be the mode of system operation?

8. What communications are anticipated for system operation?

9. What constraints on computer usage are assigned to the information system?

10. Will other organizational functions share the same hardware used by the information system?

11. Do the hardware plans anticipate future growth in the organization?

12. What are the time-phased plans for hardware implementation?

13. Is the hardware configuration modular and compatible over all levels of management?

14. What are the hardware interfaces within the organization?

15. Can software be standardized for all hardware locations?

16. What is the traffic volume between the different hardware locations?

17. Is there a requirement for vertical search from the highest level of management to the lowest level of management?

18. What are the hardware interfaces outside the organization?

19. What output display devices are being considered?

Data Collection and Data Availability Environment

In performing the error detection and correction function, the factors associated with data collection are one of the more important states of system environment. Included in the definition of data collection are the areas of data base organization and maintenance, data file updating, and data aggregation.

1. Have the system data elements been identified?

   (a) Do the data elements meet all of the user element requirements?

   (b) Have the data element sources been defined?

   (c) Have the data elements been defined at each data collection point?

(d) What is the expected volume at each collection point?

(e) What is the expected number of data collection points?

(f) Can the data collection points be grouped into geographic or functional areas?

2. What are the plans for data collection?

(a) What data collection procedures have been defined?

(b) What data collection forms have been developed?

(c) What automatic source data collection devices are contemplated?

(d) What training is required for the data collection locations?

(e) What is the expected time required to complete the data collection form?

(f) What is the frequency of source form generation per data collection location?

(g) How many copies of the form are completed by the data generator for each action?

(h) What is the disposition of the copies?

(i) What are the procedures for external data collection?

(j) In what form will external data enter the system?

(k) At what levels will external data enter the system?

(l) What is the validity (accuracy) of external data when it enters the system?

(m) What procedures were used to validate the external data?

3. What data files have been anticipated?

(a) What are the existing data files?

(b) What is the form of the existing files?

(c) What file conversion is required?

(d) What data validation of existing files are required?

(e) Determine the expected volume of the existing files.

(f) What data is contained in each existing file?

(g) What data is redundant in the existing file?

(h) Are the files integrated by having common central information?

(i) What additional data is required to update existing files?

(j) What new files are anticipated?

(k) What is the primary function of the new files?

(l) Have the sources of data for the new files been established?

(m) In what form will the new files be maintained?

(n) What system requirements dictated the form of the files?

(o) What data of the new files will be obtained outside the system? The organization?

(p) What are the formats for the data files?

(q) What are the file maintenance requirements?

(r) What is the frequency of file updating?

(s) Who has authority for updating the files?

(t) Who has responsibility for updating the files?

(u) Where are the data files located?

4. What procedures are defined for data aggregation?

(a) Where will data aggregation be performed?

(b) Are the data aggregation requirements known for each level of management?

(c) Are the aggregations at each level consistent with the decision process at that level?

(d) Are aggregated files to be maintained? For how long?

(e) Who determines the format for the aggregated files?

(f) Who determines the location of the aggregated files?

System User Environment

An important state of system environment is that of the system user. In fact, the development and operation of the system is directed by their information needs. In order to establish the necessary error detection and correction procedures to meet the user requirements, the following actions require solution.

1. Define within the current and anticipated organizational functional areas, the system users.

(a) Specify the information requirement in terms of system data elements; in terms of external data elements.

(b) Specify for each user a ranking of the data elements in order of accuracy requirements.

(c) Identify the important data element combinations for each class of users.

(d) Identify the interfaces between the information users.

(e) Define the important variables in the decision process of the information users.

2. What is the required output to the users?

(a) Have the outputs been defined for each user?

(b) Have the output formats been defined?

(c) Have the modes of output been defined?

(d) What response time is required by the users?

(e) Is there a requirement for hierarchical reports?

(f) Identify operational procedures relevant to the detection and correction function.

(g) What is the frequency of reports to each user?

(h) What is the requirement for timeliness between the occurrence of an action and its inclusion in a management report?

(i) What are the procedures for adding, changing or deleting an output product by any system user?

3. What are the required outputs for information system evaluation?

4. What system performance indices or periodical evaluation is required?

Estimating the Accuracy Loss

Accurate answers to the previous questions should provide the system designer with a fairly detailed description of where the detection and correction function can be performed and what procedures to use at the various locations. In making the final selection of locations and procedures, the system designer will rely on the cost estimates derived from the final decision array.

For example, once the data collection locations are defined as to the data collection method, the volume of data, the expected number of actions per location and procedures to be used, the cost for each location can be formulated. In a similar manner, estimates of the other locations can be performed using the guidelines of the required resources described in Chapter VI.

In providing for the final selection, care must be maintained to meet the accuracy requirements of the various system users. The process is one of iteration, where the system designer and the information users reestablish the actual requirements for data accuracy. After each iteration the system designer provides to the users the loss of detection and correction capability for various resource limitations directed at the detection and correction function. Such trade-offs will show the loss of detection and correction capability by location which can be transferred into loss of data accuracy at the locations. In addition, reductions in the requirements for data accuracy at various system levels would reduce the cost of the detection and correction function.

The ability to maintain the means for describing the loss of detection and correction capability is through the decision array. As the decision array becomes more precise through the iteration process, the system designer can evaluate the detection and correction losses at each location.

While the system designer can provide an evaluation of the loss capability, and translate this loss to data accuracy, it is the information user who must translate the loss of accuracy to their decision processes. This latter loss function between the data accuracy and the users decision process can only be estimated subjectively and only estimated by the information user. The information user may have some quantitative data concerning the dollar value of his decisions and dollar losses if wrong decisions are made

using erroneous data in his decision process. However, the final decision on the value of the accuracy loss is subjective.

The successful implementation of a totally integrated set of detection and correction procedures will provide a significant improvement in the degree of accurate data submitted at the source. That is, one of the best methods of error prevention is a good set of error detection and correction procedures. Such procedures let the generators know someone is auditing the input and that erroneous data may be returned to the generator. The form of returning erroneous data may cause either embarrassment to the organization or a personnel penalty to the generator. Probably the greatest user of such a technique is the Internal Revenue Service. In this case the question of subjective loss to the generator is quite real to those who are convicted of intentionally submitting erroneous data.

CHAPTER IX

SUMMARY AND CONCLUSIONS

Summary of the Results of the Study

The problems and decisions facing the information system designer are many. The area of error detection and correction is but one of the many problems. However, in many system designs, error detection and correction is one of the areas that does not receive its due attention. A great amount of resources are consumed on selecting the proper system hardware, system software, with elaborate mathematical models being employed to perform calculations on the data. However, little effort is expended for proper data control. Even when total costs are considered, the cost of data collection represents from one-third to one-half of the outlay for all costs other than hardware costs. In addition, this cost is not a one time cost but a cost that will continue throughout the life of the system.[1]

Such large costs, in themselves, suggest that additional resources should be expended in the area of data control. The intent of this paper has been to focus attention on the problem of data control. In particular, to focus attention on the problem of detecting and correcting input errors introduced by the human observer in recording the original data and in transferring that data into machine readable form.

---

[1] William B. Moore, The Input Problem. Lecture in WORC/TIMS Seminar Series at the Civil Service Building, Washington, D.C., (October, 1967).

The problems of erroneous data are not new to the users of information systems or to those associated with the actual computer operations. Yet, there is little formal communication between these two groups as to methods for solving these problems.

For others involved in information system development, there is little formal information on either the derivation or definition of errors. As a result of the lack of communications and any formal documentation describing the input error problem, a complete conceptual framework was developed in this paper for the error detection and correction process.

The results of this study are presented in two parts. The first set of results describe the framework that was developed as a necessary step to the formalization of the detection and correction procedures. The first set describes the framework through the development of the procedures for detection and correction, independent of where the procedures are performed. The second set of results describes the framework from the detection and correction procedures through a model for determining the range, depth and location of the detection and correction procedures.

The results of the first set include (1) a description and model of how errors are created, (2) definitions and necessary conditions required to detect and correct errors, (3) a formal description of the system locations where errors can be detected and corrected, and (4) the development of detection and correction procedures.

(1) The creation of errors

Each time data are transferred from one point to another, a sequence of events occurs. The events can be described by a decision tree, where the results of each event leave the data either correct or incorrect. For any such transfer, there are seven outcomes to the decision tree, only one of which is the correct path through the sequence. The remaining six outcomes result in input errors which are subjected to detection and correction procedures. Each

outcome of the decision tree is defined as belonging to a code set or not belonging to the code set. The code set is composed of all the acceptable variations of the event under consideration. The code set concept is used as the basis for developing the detection and correction procedures.

The development of the decision tree model is based on the sensor and how he observes and records the event. The sensor, as defined for this paper, is a human observer who either initially records the data or who transfers the data from one recording medium to other media. The same decision tree model could be used for any type of sensor. The resulting detection and correcting procedures, however, may be different.

The six error outcomes from the decision tree were classified as one of two kinds: a Type I or a Type II error. The Type I errors are limited, by definition, to errors in recording, while the Type II errors result from an initial error of observation. However, the Type II errors could result in a compound error, through an additional error in recording.

(2) Definitions and conditions for detecting and correcting errors

Before errors can be corrected, they must be detected. To detect errors, decision rules must be established defining what constitutes an error. In providing a definition for errors, certain properties were required of the data elements. These properties defined necessary, but not sufficient, conditions for errors to be detected in data elements. These properties described in the paper are repeated here. For an error to be detectable:

(a) it must not be a member of its code set. or

(b) it is not a member of the subset of other data element code sets when taken in specific combination.

Once an error has been detected, five conditions were described which enable the detected error to be corrected.

These conditions are:

(a) The code set contains error correcting digits that enable unique identification.

(b) The code, when connected to other code sets, establishes unique code set combinations that, through logical progression, are error correcting.

(c) The data element is of such a nature that bounds can be placed on detectable errors.

(d) The coded data element can be returned to the sensor for correction.

(e) The coded data element is of such a nature that statistical techniques (such as past probability estimates) can be used to determine the most likely correct code.

Two points are significant. First, even with these five conditions, not all detectable errors are correctable. Second, the complexity and cost needed to correct some detectable errors may render the effort inadvisable. The latter case is especially true for low valued data elements which require complex statistical procedures.

(3) Locations for detection and correction

This result has the unique interest of separating the information system into functional locations where error detection and correction can be performed. The locations extend from the generation of the data (the data generator location) to the final disposition of the information reports (information user location). While all the locations described may not appear in every information system, those locations that do appear will be a subset of the locations described. The primary consideration for location selection was where data was transferred in the system. The transfer was defined to include initial recording, reformatting, the transfer from one medium to another, and the aggregation of data into reports.

(4) Detection and correction procedures

A concept of detecting and correcting input data errors was formulated. The concept consists of: (1) the development of data element classes, (2) the criteria for selecting detection and correction procedures at various locations, and (3) the development of the detection and correction procedures.

The development of data element classes provided for homogenous data elements to be considered by the same set of detection or correction procedures. A significant result was the ability to separate both exact and approximate detection procedures into the data element classes consistent with the data element class description. The data classes provided a convenient method for approaching the detection and correction procedures, since it allowed for a more general description than could be provided if specific data elements were cited.

The criteria for selecting detection and correction procedures at the various locations proved significant. The criteria allowed for the procedures to be separated into non-computer-aided procedures and computer-aided procedures. This dichotomy reduced the number of variations to be considered at the locations, and provided a hierarchy for ordering the various procedures.

The developed detection and correction procedures provides for a wide range of available procedures and techniques. The significant points are that the procedures (1) are sequencial and (2) range from simple sight detection to computer-aided statistical and probabilistic correction procedures.

For the procedures to be sequential means that all of the procedures are not necessarily found or needed in any one information system. There may be several procedures for detecting the same kind of error; each procedure being more complex than the previous one. However, if the system under consideration does not require the degree of detection provided by the complex, the procedure can be eliminated from consideration.

To provide a meaningful concept of detection and correction, the procedures must reach each level of the information system. The procedures then form an audit of the input with checks and balances at the various locations. While the procedures were developed independent of the detection and correction locations, consideration was given to the resources needed to perform the procedures. Such consideration for resources resulted in classes of procedures some of which were quite simple and could be performed by the individual who actually recorded the data. Other procedures resulted in the need for elaborate data files and equipment which could not be performed efficiently by an individual. These latter procedures resulted in the complex computer-aided detection and correction procedures.

The second set of results are focused on (1) the concept of error priority, (2) the data report statistics, (3) the cost of detection and correction, and (4) a model for determining the range, depth and location of the detection and correction procedures.

(1) Error priority

The concept of error priority is defined as a ranking scheme, either ordinal or cardinal, which orders or ranks the data elements as to their importance. The higher the importance of a data element, the higher the priority of errors concerning that data element. To develop a working error priority scheme requires a great deal of interaction between the system designer and the users of the information. Therefore, this paper describes the underlying concept with attention focused on the need and use of error priority in the detection and correction procedures.

(2) Data report statistics

The users, in most computerized information systems, are removed from the actual location of report generation. To provide the user with as much information as possible, statistics concerning the data presented in the report can be very significant to the user's decision process. The paper describes several statistics that should be included in the reports at the time of generation.

Such statistics would include results of the error detection and correction procedures related to the data of the report.

In addition, uncorrectable detected errors were described. These uncorrectable detected errors are contained in data records, and consideration must be given to the disposition and use of these records. To evaluate the usefulness of these records, a truth table was developed. The truth table defined the various combinations of correct and erroneous data elements that could be used in preparing a series of reports. This concept allowed data records containing errors to be included in output reports. The concept increases the data available for reports, giving the users more complete information on which to evaluate their decisions.

The development of report statistics perform another significant role. They are used to evaluate the effectiveness of the detection and correction procedures. The evaluation is performed by monitoring selected ratios from the detection and correction process. The ratios, maintained over time, will indicate the effectiveness of the procedures. However, these ratios only consider errors that were detected and possibly corrected. As an attempt to estimate the "true" error rate, a concept of a "coefficient of detectability" was developed. The coefficient, when used in conjunction with the ratios of detected errors to total records processed would provide an estimate of the "true" error rate.

(3) The cost of error detection and correction

The cost of error detection and correction can be attributed to three basic resources: (1) personnel, (2) equipment, and (3) data collection requirement. These three resources can be used in various combinations to obtain the desired level of accuracy from the information system. In describing the relationships between these three resources, two factors are to be considered. These two factors are the data worth to accuracy relationship, and the accuracy to cost relationship.

Before meaningful cost analysis can be performed, it is necessary to obtain, from the users, a measure of the data worth associated with different levels of accuracy. The assumption behind the relationship is that all data does not have the same worth and that all data above a certain accuracy level has the same worth to the user. In fact, the user may only be able to describe his accuracy requirements in general terms. The paper focuses attention on this problem, and describes a procedure for obtaining judgments from the users.

The accuracy to cost relationship is needed to describe the range of cost that will be incurred if the accuracy levels requested are met. However, when the system has a dollar constraint, an accurate relationship will enable the system designer to better allocate the limited detection and correction resources to the higher worth data.

In describing the cost associated with the detection and correction procedures, the three major resources were discussed for seven classes of procedures. The analysis provided a description of the kind of subresource needed and in some cases a method for estimating the amount of the resource.

(4) Determining the range, depth and location of the detection and correction procedures

A model was developed for aiding the system designer in determining the range, depth and location of the procedures. The model took the form of a three-dimensional array. The location was one dimension of the array, while the range and depth of detection, and the range and depth of correction constituted the other two dimensions. The range of detection and correction is defined as the number of data element classes under investigation at a location. The depth of detection and correction is defined as the complexity of procedures used to detect or correct the data element classes.

The model is used to eliminate non-responsive variations of the range, depth and location through an

iteration process. After each iteration and elimination, the remaining alternatives are reevaluated. The reevaluation is performed in conjunction with a check list describing the states of system environment. The states of system environment describe the major functional components of the information system and the intended procedures for system operation. In particular, the states of system environment can be considered an extension of the system feasibility study.

As alternatives are eliminated from the array, a manageable and responsive subset will be reached. The subset which cannot be reduced any further by the check list, will be evaluated as to cost. The most cost-effective alternative from the subset would be selected and implemented as the system detection and correction procedures.

Conclusions

The resources being consumed in the information technology explosion have been estimated as a 40 percent share of the gross national product in 1967.[1] Yet there is a lack of information available to the users and developers of information systems on how to control errors in these systems. The knowledge concerning error content in most systems is limited to the hardware components of the system, with little regard for the error content of the input data. Most information system users either assume that all the information presented to them is error free, and use the information as presented, or have a complete lack of faith in the information and bypass the system.

---

[1]See Jacob Marschak, "Economics of Inquiring, Communicating, Deciding," Working Paper No. 134, University of California, Los Angeles, Western Management Science Institute (January, 1968). Included in this percent is all of the information knowledge including radio, television, newspapers, and management information systems.

The users are divided with regard to be value that they place in their information systems. At one extreme are managers whose security and perhaps sanity is maintained by avoiding challenges to the validity of their data bases. At the other extreme, are managers who, having found many errors within their information, are totally cynical about their automated files. Missing, both from current information system procedures and from the literature, is a rationale which provides a basis for intelligent, confident movements toward some middle ground. The provisions of that rationale, the development of a structure or philosophy of the error phenomenon is a major concern of this paper.

The middle ground, a mature attitude about input errors, can only come about through an understanding of the error generation, error detection, and error correction process. Further, management maturity in this area is very much related to the achievement of insights on the cost-effectiveness aspects of the error picture. Errors are not uniformly bad or serious. In fact there is a wide variation across the classes and types of errors in terms of their effects on the decision process. Rejection of this idea or failure to appreciate it leads, for those who assume the worst, to costly over-concern and over-investment in error protection. For those who cannot afford the insurance policy cost associated with the proper error procedures, the acceptance of errors and the occasional unfortunate decision becomes a normal characteristic of the information system.

This research has attempted to remove some of the mystery which continues to plague the input error question. A system for classifying errors by type has been developed. Attention has been paid to the kinds of errors which can be made or introduced at various levels in the data generation-data processing chain. More important, these levels and their potential use to managers and researchers alike provide a conceptual framework in which intelligent discussions concerning the error process can be formulated. The concept of data worth alone, provides a significant step forward in building an intelligent detection and correction process.

The basic means of justifying and evaluating automated information systems has been a cost displacement criterion -- mainly through reduction of clerical costs. This criterion is no longer valid. New criteria are being suggested. The basis for the new criteria are associated with the value as worth of the data in the decision process of the organization. This task of determining the value of the data cannot be left to the system analysts or senior programmers, but must be performed by the decision maker. The basic concepts needed by the decision maker for evaluating the worth of his data are covered in the paper. Not only the basic concept of data worth, but the necessary relationships between data worth, accuracy and cost.

The relationship between data worth and accuracy define the range and depth of the detection and correction procedures for the system. Attention was given to the kinds of procedures that could be employed at the locations. The locations, the range and depth of the procedures provide the user with the accuracy needed in his decision process.

To present the information user with all the possible alternative error detection and correction plans would be confusing and frustrating. Concern for this problem led to the development of a systematic procedure -- a model -- for evaluating all feasible plans. The model while preparing the information user for the complexity of solving the error problem, it gives the researcher and designer a tool for communicating between the major participants of the system. The final evaluation of the procedures to be used in the system is based on cost. Not the cost of displacement, but the cost associated with improved operations through more accurate information. Trades-off are between the cost of desired accuracy and the cost of wrong decisions from inaccurate information.

The study outlines a conceptual framework on which to build. Increased interest in information system development will require that more research be placed on the problems associated with data input. Current technolcgy is hardware

oriented. The objective is to put the data in machine readable form as soon as possible. However, the basic problem still exists -- detecting and correcting the errors of the human observer or sensor who initially records that data. The new criteria for justifying the implementation of information systems will be based, to a large extent, on the value of the information to the decision process. The value of information is the worth of the data and the worth of the data is the data accuracy problem. The major contributor to data accuracy is formal procedures for input error detection and correction.

BIBLIOGRAPHY

Books, Monographs and Pamphlets

Dixon, W. J. and Massey, F. T. Introduction to Statistical Analysis. 2nd ed. New York: McGraw-Hill Book Co., 1957.

Fischer, George Jr., et al eds., Optical Character Recognition. Washington, D.C.: Spartan Books, 1962.

General Services Administration. Source Data Automation. FPMR11.5. Washington, D.C.: Government Printing Office, 1965.

Henderson, James M. and Quandt, Richard E. Microeconomic Theory - A Mathematical Approach. New York: McGraw-Hill Book Co., 1958.

Kimball, E. W. "Malfunction and Failure Analysis," in Reliability Handbook. Edited by W. Grant Ireson. New York: McGraw-Hill Book Co., 1966.

Laden, H. N. and Gildersleeve, T. R. System Design for Computer Applications. 2nd ed. New York: John Wiley and Sons, 1967.

Luce, Duncan and Raiffa, Howard. Games and Decisions. New York: John Wiley and Sons, 1957.

Martin, James. The Design of Real-Time Computer Systems. 2nd ed. New York: McGraw-Hill Book Co., 1967.

Solomon, Herbert. ed. Mathematical Thinking in the Measurement of Behavior. Glencoe, Illinois: The Free Press of Glencoe, 1960.

Sippl, Charles J. Computer Dictionary and Handbook. Indianapolis, Indiana: Howard W. Sams and Co., 1966.

Tinker, M. A. Legibility of Print. Ames, Iowa: Iowa State University Press, 1963.

Tou, Julius T. ed. Computer and Information Sciences - II. New York: Academic Press, 1967.

Walsh, John E. Handbook of Nonparametric Statistics, II. Princeton, New Jersey: D. Van Nostrand Co., 1965.

Watson, Donald S. Price Theory and Its Use. Boston, Mass.: Haughton, Mifflin Co., 1963.

Wechsler, David. The Measurement of Adult Intelligence. Baltimore, Md.: The Williams and Wilkins Co., 1944.

Articles and Periodicals

Berkwit, George J. "Middle Managers vs The Computer." Duns Review, November, 1966.

Carlson, Gary. "Predicting Clerical Error." Datamation, February, 1966, pp. 34-36.

Conrad, R. "Errors of Immediate Memory." The British Journal of Psychology, November, 1959.

Crannell, C. W. and Parrish, J. M. "A Comparison of Immediate Memory Span for Digits, Letters and Words." The Journal of Psychology, 1957.

Deardon, John. "Can Management Information Be Automated." Harvard Business Review, March-April, 1964.

_____. "How to Organize Information Systems." Harvard Business Review, March-April, 1965.

Fasteau, Herman H., Ingram, J. Jack and Minton, George. "Control of Quality of Coding in the 1960 Census." Journal of the American Statistical Association, March, 1964, pp. 120-132.

Finerman, Aaron and Rivers, Lee. eds. "A Comprehensive Bibliography of Computing Literature, 1967." Association for Computing Machinery, 1968.

Flack, J. R. Jr. "Seven Deadly Dangers in EDP." Harvard Business Review, May-June, 1962.

Garrity, John T. "Top Management and Computer Profits." Harvard Business Review, July-August, 1963.

Lemmer, E. T. and Lockhead, G. R. "Productivity and Errors in Two Keying Tasks." J. Applied Psychology, Vol. 46, 1963.

Miller, George A. "The Magical Number Seven Plus or Minus Two, Some Limits on Our Capability for Processing Information." The Psychological Review, March, 1956.

Rabinow, Jacob. "Optical Character Recognition Today." Data Processing Magazine, January, 1966.

Smidt, Seymour. "Flexible Pricing for Computer Services." Management Science, June, 1968.

"The Information Revolution." Duns Review, September, 1966.

Reports and Proceedings

Arquette, L.; Calabi, L.; and Hartnett, W. E. A Study of Error Correction Codes, Parts I, II, III. Parke Mathematical Laboratories, Inc. Carlisle, Mass.: 1967.

Belmont, Peter A. Font Recognition. Technical Report No. RADC-TR-67-348 Sylvania Electronic Systems under Contract AF30(602)-4066, Rome Air Development Center, Griffess Air Force Base, New York: October, 1967.

Chapdelaine, P. A. Accuracy Control in Source Data Collection. Headquarters, Air Force Logistics Command. Wright-Patterson Air Force Base, Ohio: 1963.

Charnes, A. and Cooper, W. W. Data, Modeling and Decisions. Management Sciences Research Report No. 43. Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh, Pa.: June, 1965.

Churchill, Neil and Stedry, Andrew. Some Developments in Management Science and Information Systems with Respect to Measurement in Accounting. Management Sciences Research Report No. 42. Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh, Pa.: March, 1965.

Henderson, Madeline M. Bibliography on Evaluation of Information Systems. National Bureau of Standards, Center for Computer Sciences and Technology. Gaithersburg, Md.: 1967.

Kriebel, Charles H. Information Processing and Programmed Decision Systems. Management Sciences Research Report No. 69. Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh, Pa.: December, 1966.

Owsowitz, S. and Sweetland, A. Factors Affecting Coding Errors. Rand Memorandum RM-4346-PR. The Rand Corporation, Santa Monica, Calif.: 1965.

U. S. Department of Army, Research and Development Directorate, U. S. Army Materiel Command. Forecast in Depth on Information Processing Systems for the Field Army. U. S. Army Materiel Command, Washington, D.C.: June, 1965.

Watson, William A. Human Factors in the Automation of Message Subsystems. Technical Report SP-1579. System Development Corporation, Santa Monica, Calif.: July, 1964.

Unpublished Reports and Working Papers

Davis, Ruth M. Information Control in an Information System. Lecture in WORC/TIMS Seminar Series. Washington, D.C.: October, 1967. (Mimeographed)

Emery, James C. Organizational Planning and Control. University of California Extension, Los Angeles. Lectures in Management Information Systems: A Critical Appraisal. Los Angeles, Calif.: August, 1967. (Mimeographed)

Marschak, Jacob. Economic Theory of Information. Working Paper No. 118. Western Management Science Institute, University of California, Los Angeles, Los Angeles, Calif.: May, 1967.

_____. Economics of Inquiring, Communicating, Deciding. Working Paper No. 134. Western Management Science Institute, University of California, Los Angeles, Los Angeles, Calif.: January, 1968.

Lombardi, Lionello A. On-line Computation in Technology and Economics. University of California Extension, Los Angeles. Lectures in Management Information Systems: A Critical Appraisal. Los Angeles, Calif.: August, 1967. (Mimeographed)

Moore, William B. The Input Problem. Lecture in WORC/TIMS Seminar Series, Washington, D.C.: October, 1967. (Mimeographed)

U. S. Department of Navy, Bureau of Naval Personnel. Errors in Transcribing Navy Service Numbers. Pers-152, Memo 60-6. Bureau of Naval Personnel, Washington, D.C.: July, 1960.

U. S. Department of Navy, Maintenance Support Office, Maintenance Data Collection System. Validation Specifications for Naval Air Activities. Report No. MSO-AR-65-00039-02. Mechanicsburg, Pa.: August, 1967.

U. S. Department of Navy, Maintenance Support Office, Maintenance Data Collection System. Validation Specifications for Naval Shipboard Reporting. Report No. MSO-SR-65-00039-01. Mechanicsburg, Pa.: February, 1968.

U. S. Department of Navy, Maintenance Support Office. Product Confidence Report by J. A. Cohick. Technical Report 2020-409116. Mechanicsburg, Pa.: March, 1967.

U. S. Department of Navy, Maintenance Support Office. Ship Mission - Card Submission Relationships by D. W. McGraw. Technical Report 3031-413017. Mechanicsburg, Pa.: October, 1967.

U. S. Department of Navy, Maintenance Support Office. Timeliness of Ship Data Submission to Maintenance Support Office by S. W. Timmerman. Technical Report 2029-120126(S). Mechanicsburg, Pa.: May, 1967.

U. S. Department of Navy, Maintenance Support Office. The Magnitude of Shipboard Maintenance Actions Not Linked with Their Associated Parts Information by S. W. Timmerman. Technical Report 3054-409067. Mechanicsburg, Pa.: September, 1967.

U. S. Department of Navy, Maintenance Support Office. Effect of Time in Program on Data Submissions by David C. Troyer. Technical Report 3012-415096. Mechanicsburg, Pa.: December, 1966.

Willmorth, N. E. System Programming Management. Technical Report No. TM(L)222/001/01. System Development Corporation, Santa Monica, Calif.: September, 1965.

NONE
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
The George Washington University Logistics Research Project	NONE
	2b. GROUP

3. REPORT TITLE

DATA INPUT ERROR DETECTION AND CORRECTION PROCEDURES

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific

5. AUTHOR(S) (Last name, first name, initial)

Varley, Thomas C.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
2 June 1969	242	55

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-67-A-0214	T-222
b. PROJECT NO. NR 047 001	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
c.	
d.	

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research

13. ABSTRACT

This study is an examination of the input data error problem in computerized information systems. The area of concern is the detection and correction of input data errors resulting from human recording during the initial collection of the data.

The knowledge concerning error content in most systems is limited to the hardware components of the system, with little regard for the error content of the input data. Most information system users either (1) assume that all the information presented to them is error free and use the information as presented, or (2) have a complete lack of faith in the information and bypass the system.

Missing from current information system procedures and from the literature is a rationale which provides a basis for intelligent, confident movements toward some middle ground. The provisions of that rationale--the development of a structure of philosophy of the error phenomenon--are the major concern of this paper.

This research attempts to remove some of the mystery surrounding the input error problem. A system for classifying errors by type is developed; attention is paid to the kinds of errors which can be made or introduced at various levels in the data generation-data processing chain. More important, these levels and their potential use to managers and researchers alike provide a conceptual framework in which intelligent discussions concerning the error process can be formulated. The concept of data worth alone provides a significant step forward in building an intelligent detection and correction process.

to page B

DD FORM 1 JAN 64 1473

A

NONE
Security Classification

NONE
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Management Information Systems						
Error Detection						
Error Correction						
Computer-aided Procedures						
Decision Model						
Error Prevention						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

_____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

_____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

_____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

DD FORM 1 JAN 64 1473 (BACK)

NONE
Security Classification

Abstract cont'd

The basic means of justifying and evaluating automated information systems has been a cost displacement criterion--mainly through reduction of clerical costs. New criteria are being suggested. The basis for the new criteria is associated with the value or worth of the data in the decision process. The basic concepts needed by the decision maker for evaluating the worth of his data are covered in the study. The necessary relationships between data worth, accuracy, and cost are also covered.

The study develops a systematic procedure--a model--for evaluating the various detection and correction alternatives. The final evaluation of the detection and correction procedures to be used in the system is based on cost. This is not displacement cost, but cost associated with improved operations through more accurate information.

The value of information is the worth of the data, and the worth of the data is the data accuracy problem. The major contributor to data accuracy is formal procedures for input error detection and correction. This study has developed these formal procedures.